

# Entire Operations

## Konzept und Leistungsumfang

Jobsteuerung auf Mainframe- und Unix-Anlagen

**Bestellnummer: NOP411D006ALL**

Dieses Handbuch gilt für Entire Operations ab Version 4.1.1.

Hierin enthaltene Beschreibungen unterliegen Änderungen und Ergänzungen, die in nachfolgenden Update-Serien oder Neuauflagen bekanntgegeben werden.

Anmerkungen und Verbesserungsvorschläge der Leserinnen und Leser sind sehr willkommen. Bitte richten Sie Ihre Anmerkungen an:

Software AG  
Dokumentation  
Uhlandstraße 12  
64297 Darmstadt  
Telefax: 06151-92-2610

© Copyright Software AG 2002 – 2003

Alle Rechte vorbehalten

Printed in the Federal Republic of Germany

Software AG und/oder Software AG Produkte sind entweder Warenzeichen oder eingetragene Warenzeichen der Software AG. Andere hier erwähnte Produkte und Unternehmensnamen können Warenzeichen ihrer jeweiligen Eigentümer sein.

# INHALTSVERZEICHNIS

<b>WAS IST ENTIRE OPERATIONS?</b> .....	<b>1</b>
Mehr als ein Scheduler .....	3
Steuerung über Rechnergrenzen hinweg .....	3
Intelligente Prozeßsteuerung .....	5
Entire Operations Viewer .....	5
Integrationsfähig .....	6
Zusammenfassung .....	8
<b>ENTIRE OPERATIONS OBJEKTE</b> .....	<b>9</b>
Eigentümer und Benutzer-Kennungen .....	12
Job-Netzwerke .....	14
Jobs .....	16
Jobtypen .....	16
Job Attribute .....	17
Job in einer Mehrrechnerkonfiguration .....	17
Logische Bedingungen .....	18
Mailboxen .....	22
Ressourcen .....	25
Kalender .....	26
Zeitpläne .....	26
Symbol-Tabellen und Symbole .....	27
Laufnummern .....	28
Laufnummern-Bereich .....	28
Reservierte Laufnummern .....	28

<b>ENTIRE OPERATIONS KOMPONENTEN .....</b>	<b>29</b>
Die Master-Datenbank .....	30
Die Externen Job-Steueranweisungen .....	30
Die Aktive Datenbank .....	31
Der Monitor .....	32
Das Betriebssystem .....	32
Die Benutzerschnittstelle .....	33
Einheitlicher Maskenaufbau .....	34
Zeilenbefehle .....	34
Direktbefehle .....	34
PF-Tasten .....	34
Fenster-Technik .....	35
Die Online-Hilfe .....	36
Deutsch und Englisch .....	37
Farbunterstützung .....	37
Die Log-Funktionalität .....	38
Die Report-Funktionalität .....	38
Der Editor .....	39

<b>ENTIRE OPERATIONS FUNKTIONEN .....</b>	<b>41</b>
Das Hauptmenü .....	42
Verwaltung der Job-Netzwerke .....	43
Zeitplanung für Job-Netzwerke .....	44
Job-Verwaltung .....	46
Job-Zeitplanung .....	47
Kalender-Definition .....	48
Verwaltung logischer Bedingungen .....	49
Eingabe-Bedingungen .....	50
Ausgabe-Bedingungen .....	52
Jobende-Prüfung .....	53
Standard-Prüfungen .....	54
Jobende-Aktionen .....	55
Benutzung von Ressourcen .....	56
Dynamische Generierung von Job-Steueranweisungen (JCL) .....	57
Beispiel 1: Dynamische Job-Steueranweisungen in einer OS/390-Umgebung: .....	58
Beispiel 2: Dynamische Job-Steueranweisungen in einer BS2000/OSD-Umgebung: .....	59
Beispiel 3: Dynamische Job-Steueranweisungen in einer UNIX-Umgebung: .....	62
System-Objekte editieren .....	64
Berichte generieren .....	65
 <b>ANHANG .....</b>	 <b>67</b>



## WAS IST ENTIRE OPERATIONS?

Entire Operations ist im Kern ein Software AG System, das den selbstgesteuerten, automatisierten Ablauf von Datenverarbeitungsprozessen (Job-Netzwerke) ermöglicht. Es bietet umfassende Funktionalität, um beinahe beliebige Arten von Hintergrundverarbeitung zu definieren. Die Prozeßverarbeitung selbst wird von Entire Operations initiiert und überwacht. Dabei wendet Entire Operations all die Regeln an, die ihm vorher in einer komfortablen Benutzeroberfläche beigebracht worden sind. Diese Benutzeroberfläche besitzt alle Merkmale einer modernen, aufgabenorientierten Benutzerführung:

- objektorientierte Menüs und Direktkommandos zur Navigation im System
- Verwendung der Fenster-Technik für Hilfetexte und Auswahlfunktionen
- online verfügbares Tutorial
- Verwendung von PF-Tasten zum schnellen Funktionsaufruf
- Grafische Schnittstelle für Windows

Natürlich ist Entire Operations in Deutsch und Englisch verfügbar, andere Landessprachen können je nach Bedarf unterstützt werden.

Zum Einsatz von Entire Operations sind keine Modifikationen am Betriebssystem oder einer der verwendeten Subsysteme erforderlich. Bestehende Job-Steueranweisungen (JCL) können unverändert unter der Kontrolle von Entire Operations eingesetzt werden, dies ermöglicht einen sanften Übergang von bestehender Produktionssteuerung hin zu automatisierter Ablaufsteuerung:

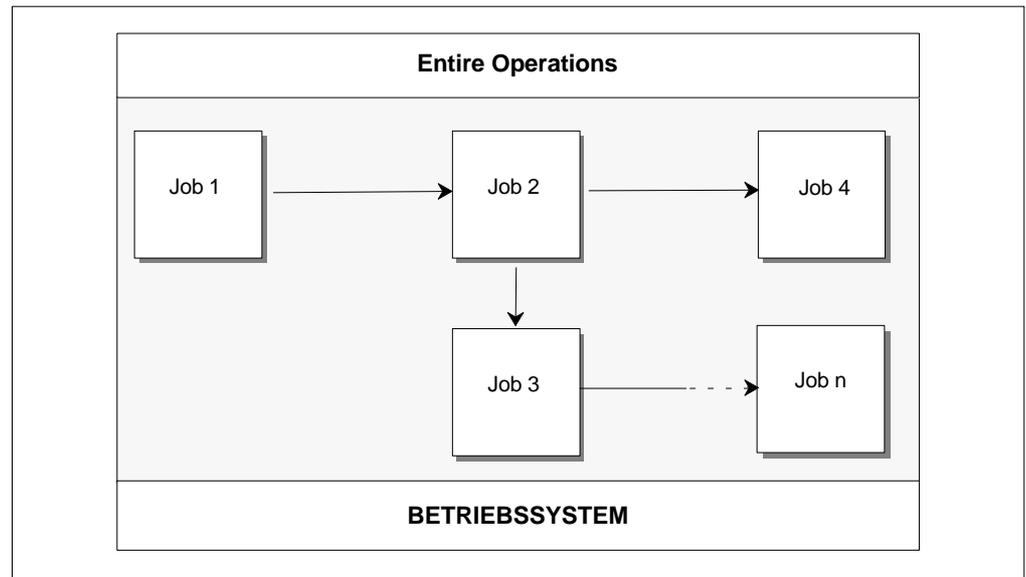


Abbildung 1-1: Entire Operations steuert und überwacht Job-Netzwerke

Bestehende Sicherheitssysteme wie RACF, ACF2, CA-TOP-SECRET oder SECOS werden unterstützt, das heißt, dort definierte Sicherheitsprofile finden auch bei der Arbeit mit Entire Operations Anwendung.

Zur Ausführung von Batch-Jobs und Prozeduren benutzt Entire Operations wohldefinierte Schnittstellen zu den vorhandenen Spooling-Systemen bzw. funktional äquivalenten Teilen des Betriebssystems selbst.

## Mehr als ein Scheduler

---

Die vorangegangenen Abschnitte beschreiben die Kernfunktionen, die mehr oder weniger wohl jedes Scheduling-System bietet. **Entire Operations aber kann mehr!**

### Steuerung über Rechengrenzen hinweg

- **Großrechnerumgebungen:**  
Entire Operations kann in allen TP-Umgebungen der Betriebssysteme OS/390, VSE/ESA und BS2000/OSD eingesetzt werden, es unterstützt insbesondere Com-plete, CICS, TSO, IMS, UTM und TIAM.
- **UNIX:**  
Entire Operations steuert Produktionsprozesse auf UNIX-Anlagen (HP-UX, SINIX RM, AIX), zum Beispiel, in Verbindung mit einer Großrechnerumgebung. Sie können aber auch Ihre Großrechner-Produktion von einem zentralen UNIX-Knoten kontrollieren oder aber Entire Operations benutzen, um einen Verbund von UNIX-Rechnern zu steuern.

Wird Entire Operations in einer Mehrrechnerkonfiguration eingesetzt, bei der die einzelnen Rechnerknoten mit Entire Net-Work, der Kommunikationssoftware der Software AG verbunden sind, so kann Entire Operations auch hier Hintergrundprozesse überwachen und steuern. In einer solchen verteilten Umgebung können selbst einzelne Job-Netzwerke unter der Kontrolle von Entire Operations aus Verarbeitungsschritten bestehen, die auf verschiedenen Rechnerknoten ausgeführt werden:

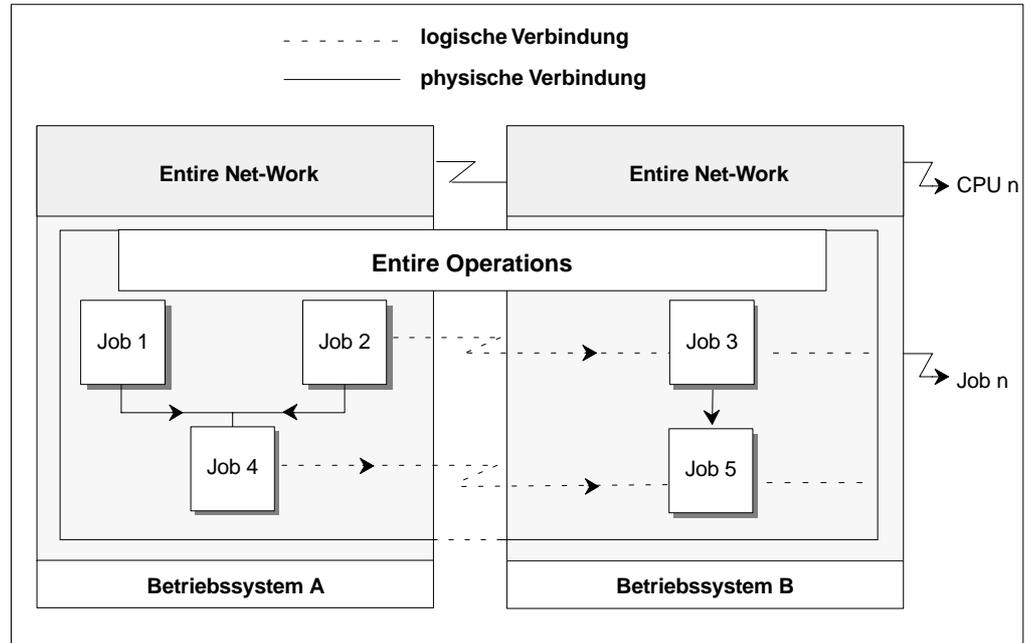


Abbildung 1-2: Entire Operations in einer Mehrrechner-Umgebung

Das Monitor-Programm von Entire Operations sorgt für die Verteilung und dezentrale Ausführung der Prozesse. Dessen ungeachtet bietet Entire Operations die Möglichkeit, diese verteilte Verarbeitung auch zentral zu Überwachen und zu steuern ("single-point-of-management").

Sie können mit Entire Operations sogar eine dynamische Verteilung der Arbeitslast ("workload balancing") vornehmen: anhand von aktuellen Systemsituationen kann während der Verarbeitung entschieden werden, auf welchem Rechnerknoten der nächste Verarbeitungsschritt ausgeführt werden soll.

## Intelligente Prozeßsteuerung

Jede Prozessautomatisierung hat die Notwendigkeit, die zukünftige Verarbeitung vorauszuplanen, um sie in der Sprache des Automationstools abzubilden. Leider unterscheidet sich manchmal die spätere Realität von der heutigen Planung. Nun kann man für solche Fälle natürlich Vorkehrungen treffen, die eine große Reihe von Ausnahmesituationen abdecken. Und natürlich bietet Ihnen Entire Operations hierfür vielfältige Möglichkeiten, um auf solche Fehler gezielt reagieren zu können.

Aber dieser Prozeß der gedanklichen Vorwegnahme möglicher Systemsituationen läßt sich nicht beliebig weit ausdehnen: jeder neue Verarbeitungszweig besonders in umfangreichen Netzwerk-Topologien bedeutet eine Komplexitätssteigerung, die weder erwünscht noch notwendigerweise hilfreich sein dürfte.

Hierfür bietet Entire Operations die Möglichkeit an, die Verarbeitungsschritte selbst variabel zu gestalten. So können zum Beispiel die auszuführenden Job-Steueranweisungen (JCL) dynamisch anhand aktueller Systemgegebenheiten (Plattenplatz, Inhalt von Warteschlangen im System, Vorhandensein von Dateien usw.) aufgebaut werden. Die Job-Steueranweisungen "lernen" also aus der augenblicklichen Situation und passen sich ihr an. Hierbei ist es sehr hilfreich, daß als Werkzeug die 4GL Natural zur Verfügung steht. Diese ist einerseits ausdrucksstark genug, um alle denkbaren Entscheidungskriterien abzubilden, andererseits stellt sie aber auch die benötigten Daten aufgrund vielfältiger Schnittstellen zu allen gängigen Datenhaltungs- und Betriebssystemen – auch in heterogenen Netzwerkeumgebungen – zur Verfügung.

So ist es dann also nicht mehr nötig, alle möglichen Problemsituationen vorwegzunehmen und bereits im Vorfeld feste Verarbeitungsschritte hierfür zu finden, sondern es müssen nur die Strategien (= Programme) zur Problemerkennung und -behebung vorgegeben werden.

## Entire Operations Viewer

Sie können Diagramme Ihrer Entire Operations Netzwerke visualisieren und analysieren mit seiner Windows-Komponente Entire Operations Viewer: entladen Sie einfach die Job-Netzwerk-Daten von Ihrem Hauptrechner zu Ihrem PC. Dazu verwenden Sie Entire Connection oder ein beliebiges Datei-Übertragungs-Programm. Mit dem Entire Operations Viewer können Sie nun in Ihren Netzwerk-Diagrammen blättern, sie vergrößern und auch drucken.

## Integrationsfähig

Wer hat sich nicht schon darüber geärgert, daß man zur Erledigung seiner Arbeit mehrere Anwendungen oder Programmsysteme braucht. Hier waren dann oftmals langwierige (und langweilige) An- und Abmeldevorgänge vonnöten oder ein übergeordneter Session-Manager notwendig (was den Ressourcenverbrauch nicht unbedingt verminderte). Entire Operations dagegen bietet die Möglichkeit, beliebige andere Applikationen in sein Hauptmenu zu integrieren. Sofern weitere Produkte der Entire System Management Linie, nämlich:

- Entire Output Management
  - Entire Event Management
  - Entire Distribution Services
  - Entire System Server
- oder aber
- Natural ISPF

vorhanden sind, erweitert sich das Einstiegsmenü von Entire Operations automatisch.

Neben dieser eher programmtechnischen Integrationsfähigkeit bietet Entire Operations die Chance, alldiejenigen Personen am Produktionsprozeß aktiv zu beteiligen, die das organisatorisch schon immer hätten tun sollen, aufgrund fehlender technischer Möglichkeiten aber bisher nicht konnten. Gemeint ist hier die Möglichkeit, über ein integriertes Mailbox-Konzept die reine Hintergrundverarbeitung eines Schedulers mit Online-Benutzern zu verbinden. Zu selbstdefinierten Zeitpunkten während der Verarbeitung von Hintergrundprozessen können Meldungen oder aber Eingabeaufforderungen an solche Mailboxen geschickt werden. Dies bewirkt einerseits, daß die Verarbeitung an diesem Punkt stoppt und daß andererseits alle die mit dieser Mailbox verbundenen Benutzer informiert werden. Diese können dann entsprechend reagieren, indem sie die geforderten manuellen Tätigkeiten ausführen (zum Beispiel, Papier im Drucker nachfüllen) oder Variablenwerte eingeben, die für die weitere Verarbeitung benötigt werden. Nach Bestätigung der Eingabe wird die Verarbeitung fortgesetzt.

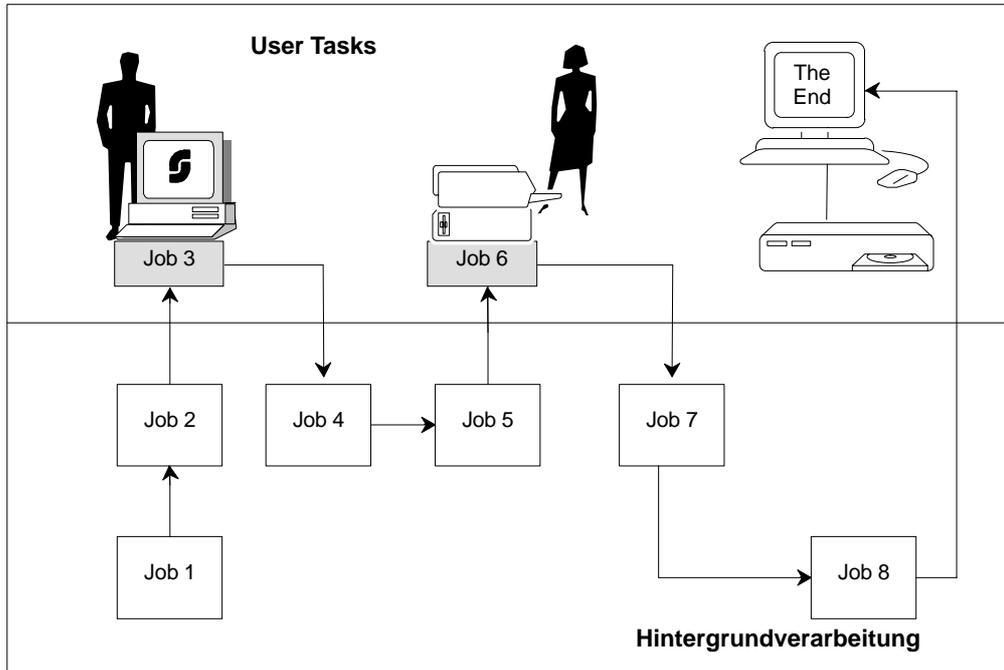


Abbildung 1-3: Integration von Fachpersonal in die automatisierte Ablaufsteuerung

Mit dieser Technik kann z.B. erreicht werden, daß fachbereichsspezifische Informationen auch von Mitgliedern der Fachabteilung eingegeben werden können, und daß trotzdem die Steuerung der gesamten Hintergrundverarbeitung unter zentraler Kontrolle bleibt.

## Zusammenfassung

---

Mit dem Einsatz von Entire Operations als automatisiertes Ablaufsteuerungssystem sichern sie sich folgende wichtige Vorteile:

- Transparente Unterstützung verschiedener Computer-Knoten, und dies sogar in heterogenen Umgebungen mit OS/390-, BS2000/OSD-, VSE/ESA- und UNIX-Plattformen;
- Verfügbarkeit in allen Host- und UNIX-Umgebungen, die von Natural unterstützt werden, zum Beispiel Com-plete, CICS, TSO, IMS, CMS, UTM and TIAM;
- Einfache Bedienung durch Menüs, Fenster-Technik, cursor-sensitive Hilfe und ein online Tutorial-System;
- Verfügbarkeit in Deutsch und Englisch;
- Existierende Job-Steueranweisungen laufen unverändert unter der Kontrolle von Entire Operations;
- Keine Modifikationen am Betriebssystem;
- Einsatz dynamisch aufgebauter Job-Steueranweisungen, bzw. Prozeduren, dadurch Zugriff auf jegliche Information des Betriebssystems oder auf ein verfügbares Datenbanksystem zur Ausführungszeit der Jobnetze;
- Integration von online Anwendern in den Batch-Produktionsprozeß durch das Konzept der Mailboxen;
- Offene Schnittstelle für Benutzer-Applikationen: Informationen aus Entire Operations können in jedes Produktionsprogramm integriert werden, Anwender können andererseits ihren Input für die tägliche Produktion eingeben, auch für einige Tage im voraus.

## ENTIRE OPERATIONS OBJEKTE

Die Benutzung von Entire Operations umfaßt u.a. die Definition und Verwaltung bestimmter Objekte (Entitäten), welche in den folgenden Abschnitten näher beschrieben werden. Hierzu gehören:

Eigentümer und Benutzer-Kennungen;

Netzwerke;

Jobs;

Logische Bedingungen;

Mailboxen;

Ressourcen;

Kalender;

Zeitpläne;

Symbol-Tabellen und Symbole.

Die folgende Abbildung veranschaulicht diese Objekte und ihre Beziehungen untereinander.

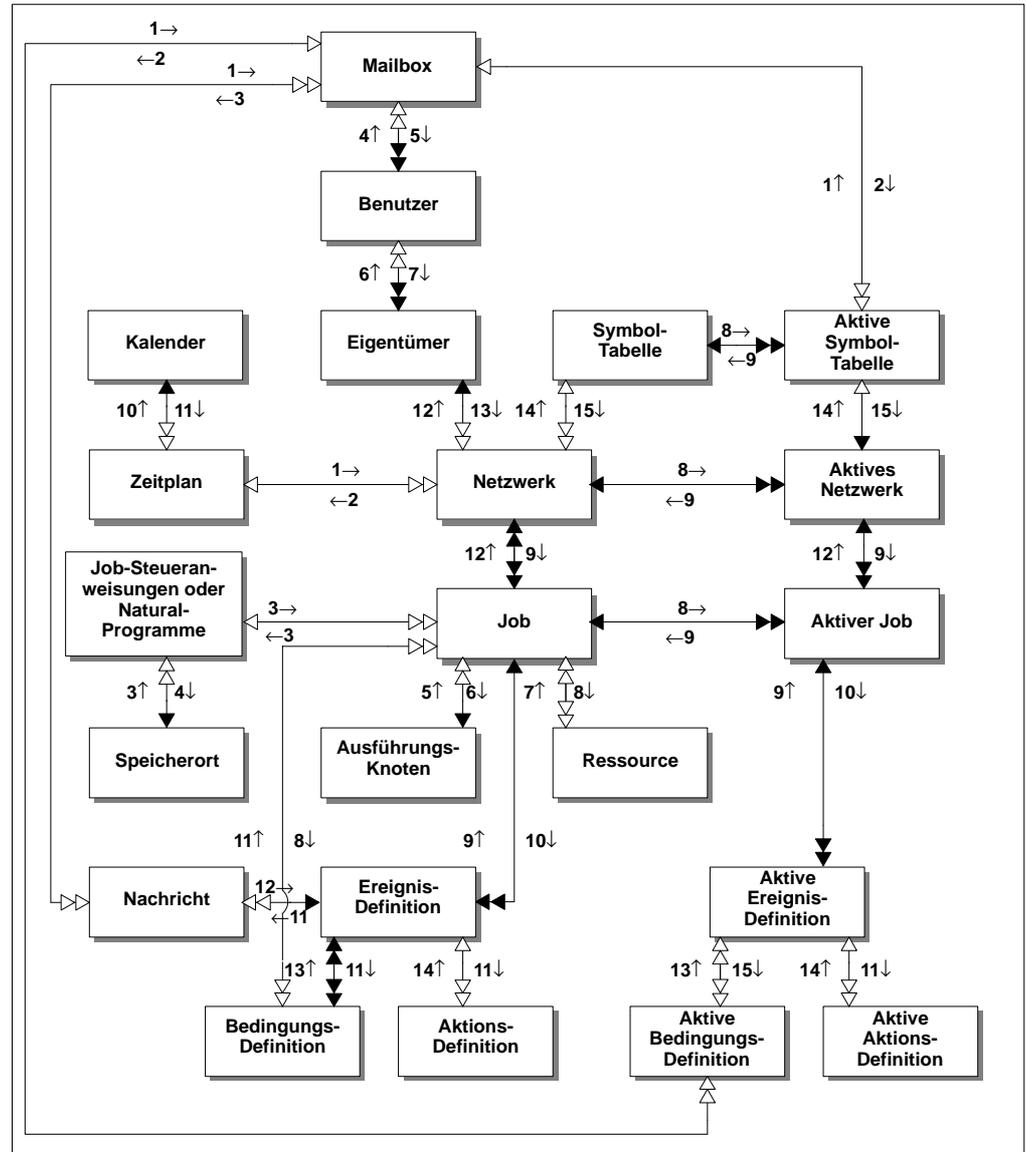


Abbildung 2-1: Entire Operations Objekte

1 = wird geschickt an	1 = steuert
2 = beinhaltet Aufforderung zur	2 = wird gesteuert basierend auf
3 = beinhaltet	3 = wird ausgeführt als
4 = benutzt / kann benutzen	4 = ist gespeichert in
5 = enthält Nachrichten für	5 = ist Plattform für
6 = autorisiert	6 = läuft auf
7 = kann wählen	7 = ist Vorbedingung für
8 = verleiht Eigenschaften an	8 = hängt ab von
9 = besteht aus	9 = bestimmt Ergebnis von
10 = basiert auf	10 = wird geprüft auf
11 = ist Basis für	11 = stößt an
12 = gehört	12 = wird gesetzt von
13 = besitzt	13 = wird (neu) gesetzt von
14 = erhält Parameter von	14 = wird ausgeführt bedingt durch
15 = enthält Werte für	15 = setzt

←	genau eins
◁	höchstens eins
◀◀	keins od. beliebig viele
◀◀	mindestens eins

## Eigentümer und Benutzer-Kennungen

Jeder Benutzer von Entire Operations erhält eine eindeutige **Benutzer-Kennung**, die für Sicherheitsprüfungen, Profilierung, Benachrichtigungen und Protokollierung herangezogen wird. Jede Benutzer-Kennung ist mit einem Benutzerprofil verknüpft, das seine Autorisierungen regelt. Benutzerprofile können von dazu berechtigten Benutzern (wie z.B. dem Systemverwalter) verändert werden.

**Eigentümer**-Kennungen werden verwendet, um einerseits Sicherheitsaspekte zu implementieren und andererseits Benutzer gruppieren zu können. Letzteres dient dazu, Verbindungen zu Job-Netzwerken zu vereinfachen. Ein Benutzer kann autorisiert werden, mehrere Eigentümer-Kennungen zu verwenden: ein Wechsel der Eigentümer-Kennung bedeutet dann, eine andere Menge von Job-Netzwerken zu selektieren, die in einem 2. Schritt verwaltet werden können.

Die folgende Abbildung zeigt ein Beispiel für die Verbindungen zwischen Benutzern, Eigentümern und Job-Netzwerken:

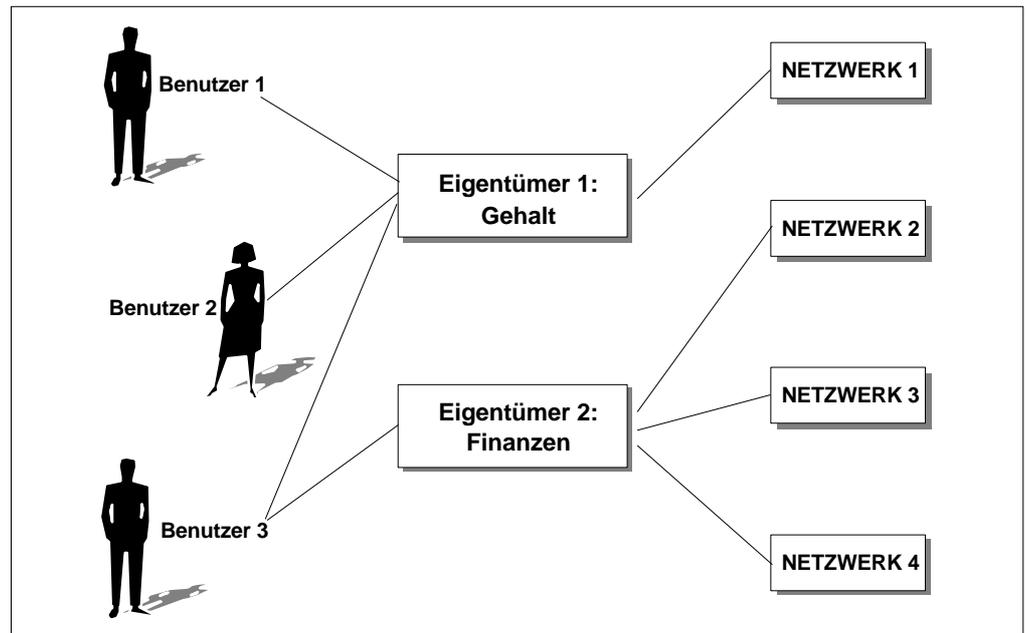


Abbildung 2-2: Benutzer, Eigentümer und Job-Netzwerke

Jeder der drei dargestellten Benutzer kann alle Netzwerke verwalten, die dem Eigentümer *Gehalt* gehören, darüberhinaus ist *Benutzer 3* für alle Netzwerke autorisiert, die dem Eigentümer *Finanzen* zugeordnet sind.

## Job-Netzwerke

Innerhalb Entire Operations umfaßt ein **Job-Netzwerk** eine Gruppe von Jobs, Tasks, Prozeduren oder allgemein Prozessen, die in Verbindung stehen können (aber nicht müssen) und die entsprechend eines Netzwerk-Zeitplans für die Abarbeitung aktiviert werden. Demnach kann ein Job-Netzwerk eine beliebige Einheit innerhalb des Produktionsprozesses darstellen.

So können in einen Job-Netzwerk auch manuelle Tätigkeiten eingebaut werden, die zu bestimmten Zeitpunkten oder in wohldefinierten Situationen vom Personal im Rechenzentrum oder aber der Fachabteilung ausgeführt werden sollen. Die folgende Abbildung veranschaulicht dies am Beispiel eines Job-Netzwerkes, so wie es in einer Lohnbuchhaltung eingesetzt sein könnte: es umfaßt die automatische Erstellung von Rechnungsbelegen und Abstimmlisten:

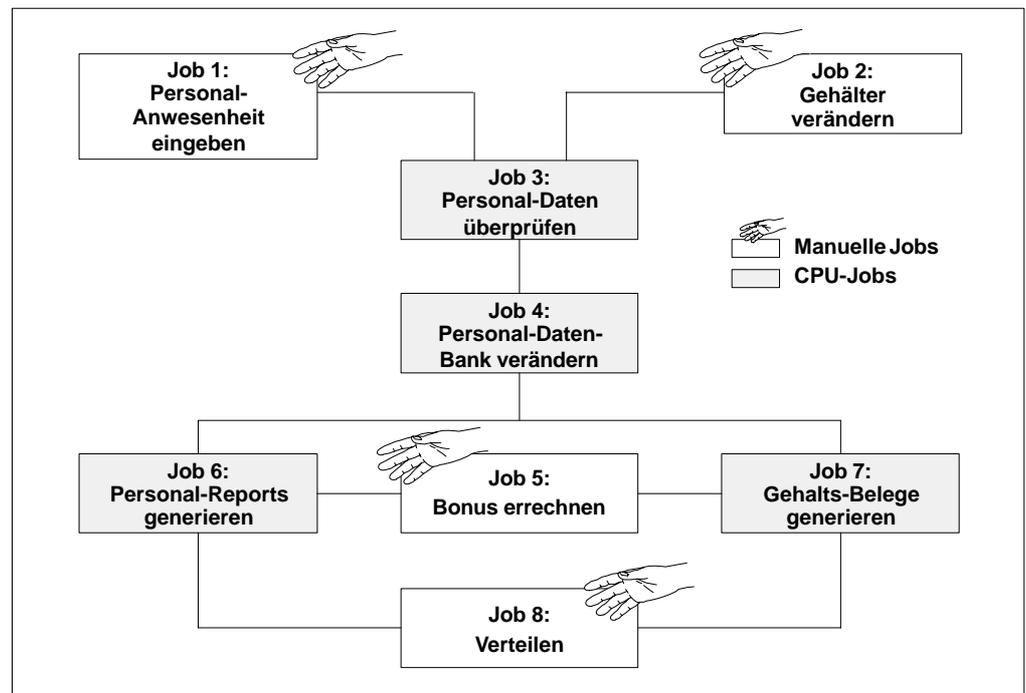


Abbildung 2-3: Ein Job-Netzwerk

Im Normalfall besteht ein Job-Netzwerk aus einer Folge von Jobs, die durch bestimmte Abhängigkeiten untereinander verbunden sind (z.B.: *Wenn Job 1 OK geendet hat , starte Job 2*). Diese Abhängigkeiten werden durch sogenannte 'logische Bedingungen' dargestellt.

Ein Job-Netzwerk ist die kleinste Einheit, die von Entire Operations automatisch aktiviert werden kann. Durch die Verwendung von Kalendern können Job-Netzwerke automatisch vom Entire Operations Monitor-Programm aktiviert werden. Manuelle Aktivierungen können von jedem dafür autorisierten Benutzer vorgenommen werden.

Mehrere aktive Kopien (=Aktivierungen) des gleichen Netzwerkes können zu einem Zeitpunkt parallel verarbeitet werden, weil Entire Operations jede dieser Kopien durch eine eindeutige Laufnummer identifiziert. Diese Laufnummer wird jedem Netzwerk bei dessen Aktivierung automatisch zugeordnet.

## Jobs

---

Der **Job** stellt einen der zentralen Objekttypen innerhalb des Entire Operations Systems dar. Ein Job kann entweder eine computer-basierte Aufgabe (CPU Job) oder eine manuelle Tätigkeit sein, die von einem Benutzer durchgeführt wird.

### Jobtypen

Entire Operations kennt folgende **Typen** von CPU Jobs:

Standard Jobs des Betriebssystems (OS/390, VSE/ESA, BS2000/OSD);

Standard Shell-Prozeduren des UNIX-Betriebssystems;

Gestartete Tasks (OS/390);

Natural Programme;

Natural MACRO Jobs (aufgebaut mit dynamischen Job-Steueranweisungen);

Zyklische Jobs (z.B. für Überwachungsfunktionen).

- Datei-Generierung.

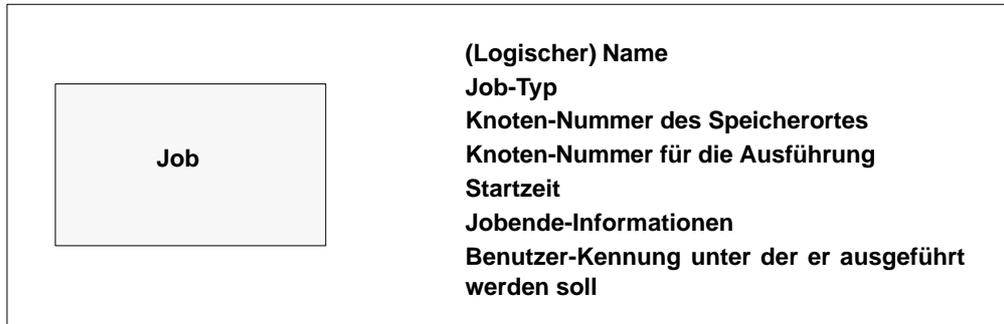
Weitere Informationen über Job-Typen entnehmen Sie dem Abschnitt **Jobtypen** im Kapitel **Job-Verwaltung** in Ihrer *Entire Operations Benutzerdokumentation*.

Daneben gibt es für nicht-CPU basierte Jobs noch den Typ des

- Dummy Jobs, der es erlaubt, Zeitfenster für nicht-CPU-basierte Jobs darzustellen oder beliebige Boolesche Verknüpfungen von Einzelbedingungen zu realisieren.

## Job Attribute

Jeder Job eines Netzwerks wird durch eine Reihe von *Attributen* identifiziert:



Ein solcher Job kann auch in *mehreren Job-Netzwerken* enthalten sein.

## Job in einer Mehrrechnerkonfiguration

Wird Entire Operations in einer *Mehrrechnerkonfiguration* eingesetzt, kann der Speicherort eines Jobs (d.h. seines Inhalts) vom Ort seiner Ausführung abweichen: zur Ausführungszeit liest Entire Operations die Job-Informationen vom Speicherort und führt den Job auf dem Zielknoten aus.

Jobs in Netzwerken können durch sogenannte 'logische Bedingungen' untereinander verbunden sein.

## Logische Bedingungen

---

Die Verwendung **logischer Bedingungen** ist *das zentrale Konzept* von Entire Operations. Logische Bedingungen werden benutzt, um Abhängigkeiten zwischen Jobs oder Job-Netzwerken zu beschreiben. Solche logischen Bedingungen können entweder durch CPU-basierte Ereignisse oder aber durch manuelle Eingaben gesetzt werden. Diese Ereignisse müssen also eingetreten sein, bevor Entire Operations mit weiteren Schritten fortfahren kann.

Sobald ein Job-Netzwerk aktiviert wird, erhält jede logische Bedingung eine Laufnummer. Mit ihrer Hilfe kann Entire Operations zwischen gleichen Ereignissen in unterschiedlichen Netzwerk-Aktivierungen unterscheiden.

Logische Bedingungen können auf folgende Weise verwendet werden:

als **Eingabe-Bedingungen**;

- als **Ausgabe-Bedingungen**.

Die folgende Abbildung veranschaulicht das Prinzip der Eingabe- und Ausgabe-Bedingungen im Verhältnis zu einem Job:

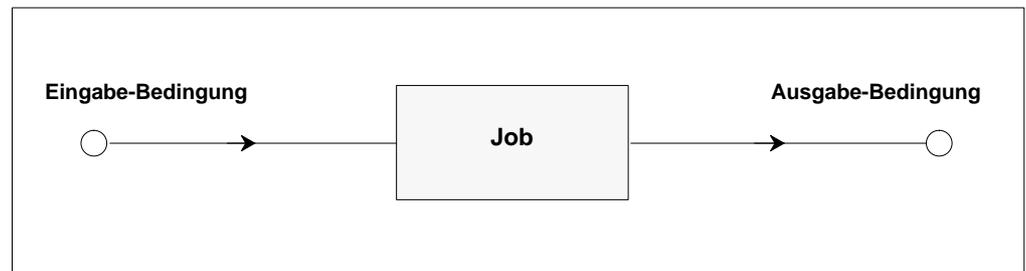


Abbildung 2-4: Eingabe- und Ausgabe-Bedingungen

Alle Eingabe-Bedingungen müssen erfüllt sein, bevor ein Job tatsächlich ausgeführt werden kann (Voraussetzung). Sie können eine beliebige Anzahl von Eingabe-Bedingungen für einen Job definieren.

Eine Ausgabe-Bedingung dagegen kann gesetzt oder aber zurückgesetzt werden, entsprechend des Ergebnisses vordefinierter Ereignisse. Diese Ereignisse sind entweder automatisch durch Entire Operations gegeben, sie können aber auch vom Benutzer definiert werden. Als Bestandteil der Jobende-Untersuchung überprüft Entire Operations das Vorhandensein solcher Ereignisse. Für jedes dieser Ereignisse können dann auf Job- oder aber Job-Step-Ebene mehrere Ausgabe-Bedingungen gesetzt oder zurückgesetzt werden.

Jobs in Job-Netzwerken werden verknüpft, indem die Ausgabe-Bedingung des einen Jobs als Eingabe-Bedingung des nächsten Jobs definiert wird. Dies wird in folgender Abbildung deutlich:

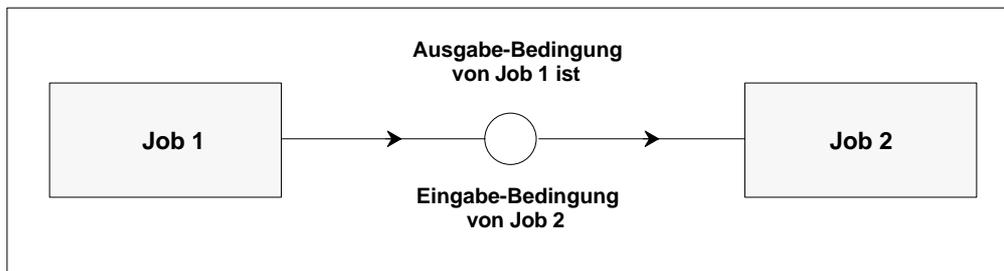


Abbildung 2-5: Jobs verknüpft durch Eingabe- und Ausgabe-Bedingungen

Tritt nun ein bestimmtes Ereignis als Ergebnis von **Job 1** ein, so setzt dies die entsprechende Bedingung und signalisiert Entire Operations, **Job 2** zu starten.

Eine beliebige Anzahl logischer Bedingungen kann für einen Job (oder aber Job-Step) definiert werden. Sie können sogar Jobs miteinander verknüpfen, die unterschiedlichen Job-Netzwerken angehören oder die auf verschiedenen Rechnerknoten ausgeführt werden.

Die folgende Abbildung zeigt Job-Abhängigkeiten am Beispiel des Netzwerks für die Lohnbuchhaltung:

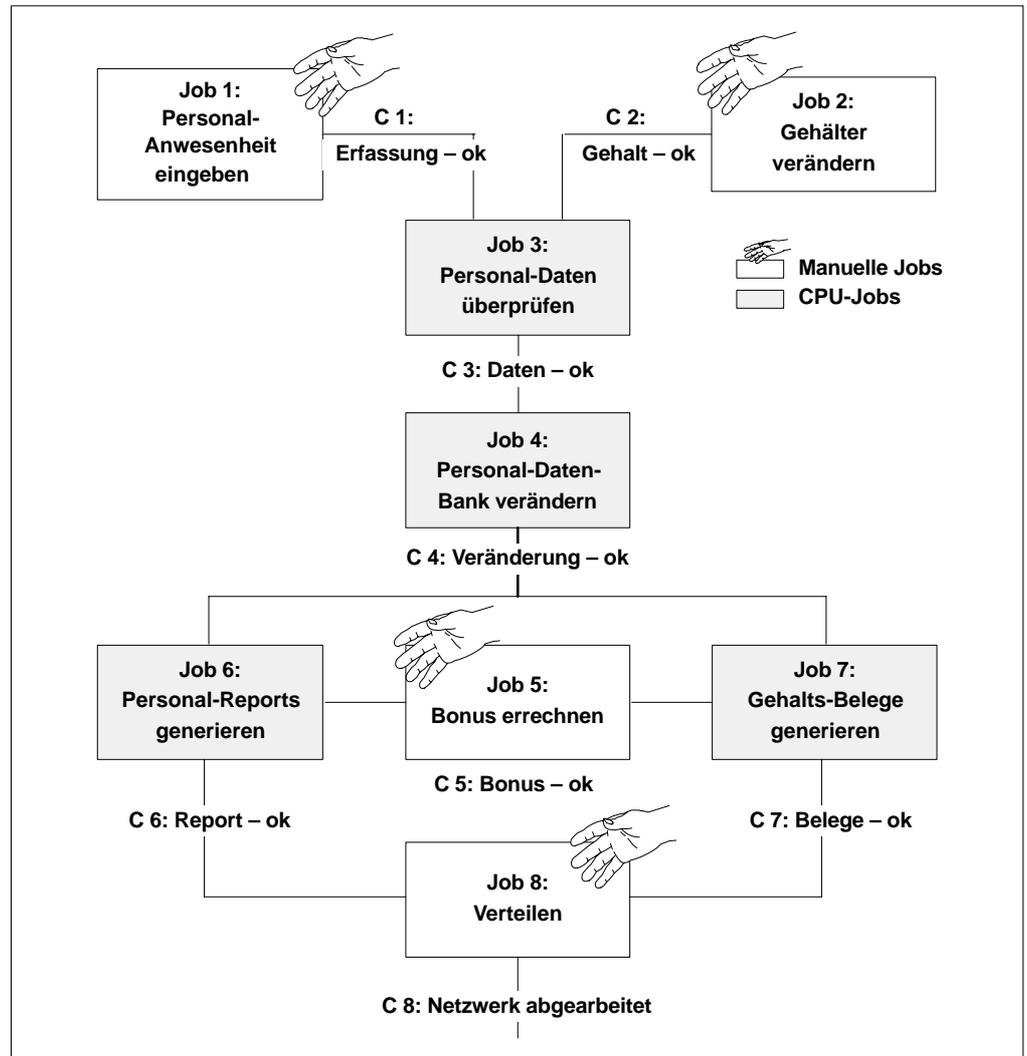


Abbildung 2-6: Job-Netzwerk mit logischen Bedingungen

Die folgende Tabelle gibt einen Überblick über die Job-Abhängigkeiten (logische Bedingungen), so wie sie die Jobs in Abbildung 2-6 (vorige Seite) verknüpfen:

Job Nummer	Eingabe-Bedingung	Ausgabe-Bedingung
Job 1		C 1: Erfassung – ok
Job 2		C 2: Gehalt – ok
Job 3	C 1: Erfassung – ok	
	C 2: Gehalt – ok	C 3: Daten – ok
Job 4	C 3: Daten – ok	C 4: Veränderung – ok
Job 5		C 5: Bonus – ok
Job 6	C 4: Veränderung – ok	
	C 5: Bonus – ok	C 6: Report – ok
Job 7	C 5: Bonus – ok	
	C 4: Veränderung – ok	C 7: Belege – ok
Job 8	C 6: Report – ok	
	C 7: Belege – ok	C8: Netzwerk abgearbeitet

Abbildung 2-7: Jobs verknüpft durch Eingabe- und Ausgabebedingungen

Zum Beispiel wird Entire Operations den **Job 6** (Personal-Reports generieren) solange nicht starten, bis die Eingabe-Bedingungen **C4** und **C5** erfüllt sind. Diese sind auch als Ausgabe-Bedingungen für die **Jobs 4** bzw. **5** definiert.

Ein solcher Jobfluß ist völlig unabhängig von den Betriebssystem-Plattformen, auf denen die einzelnen Verarbeitungsschritte ablaufen.

## Mailboxen

---

Innerhalb Entire Operations dienen **Mailboxen** dazu, mit Netzwerken verknüpfte Meldungen und Aufforderungen an Benutzer und/oder Gruppen von Benutzern zu empfangen. Diese Meldungen können benutzt werden, um Benutzer über den aktuellen Status eines Netzwerks zu informieren oder aber die Eingabe von Daten anzufordern, die für die weitere Ausführung benötigt werden.

Unter solchen Meldungen bzw. Aufforderungen werden zusammengefaßt:

Benutzerdefinierte Meldungen;

Systemmeldungen des Entire Operations Monitor-Programms;

logische Bedingungen betreffende Eingabe-Aufforderungen, die von bestimmten Benutzern bestätigt werden müssen. Diese Bedingungen sind als von einer manuellen Tätigkeit abhängig definiert worden;

- Aufforderungen für noch offene Symbol-Eingaben, die zu zukünftigen Netzwerk-Aktivierungen gehören.

Mit Hilfe des Konzepts der Mailboxen kann Entire Operations nicht-CPU-basierte Prozesse auf die gleiche Weise behandeln wie CPU-basierte:

sie können von logischen Bedingungen abhängig gemacht werden und selbst solche setzen;

- durch die Zuordnung einer Mailbox zu diesen logischen Bedingungen kann festgelegt werden, wer über diese Bedingung informiert werden soll.

Jeder Benutzer, der mit einer Mailbox verknüpft ist, wird deren Liste der ausstehenden Meldungen und Aufforderungen erhalten. Hierzu muß er lediglich den Direktbefehl **MAIL** eingeben. Durch einen einzigen Tastendruck kann der Benutzer dann die Bedingung als erfüllt markieren oder andere der geforderten Tätigkeiten durchführen, wie z.B. die Werteingabe für Symbole. Danach verschwindet diese Meldung von der Liste, das Entire Operations Monitor-Programm wird über die erfolgte Bestätigung automatisch informiert und wird alldiejenigen Jobs anstoßen, die auf diese Bedingung gewartet haben.

Die folgende Abbildung veranschaulicht das Konzept der Mailboxen am Beispiel des Netzwerks für die Lohnbuchhaltung:

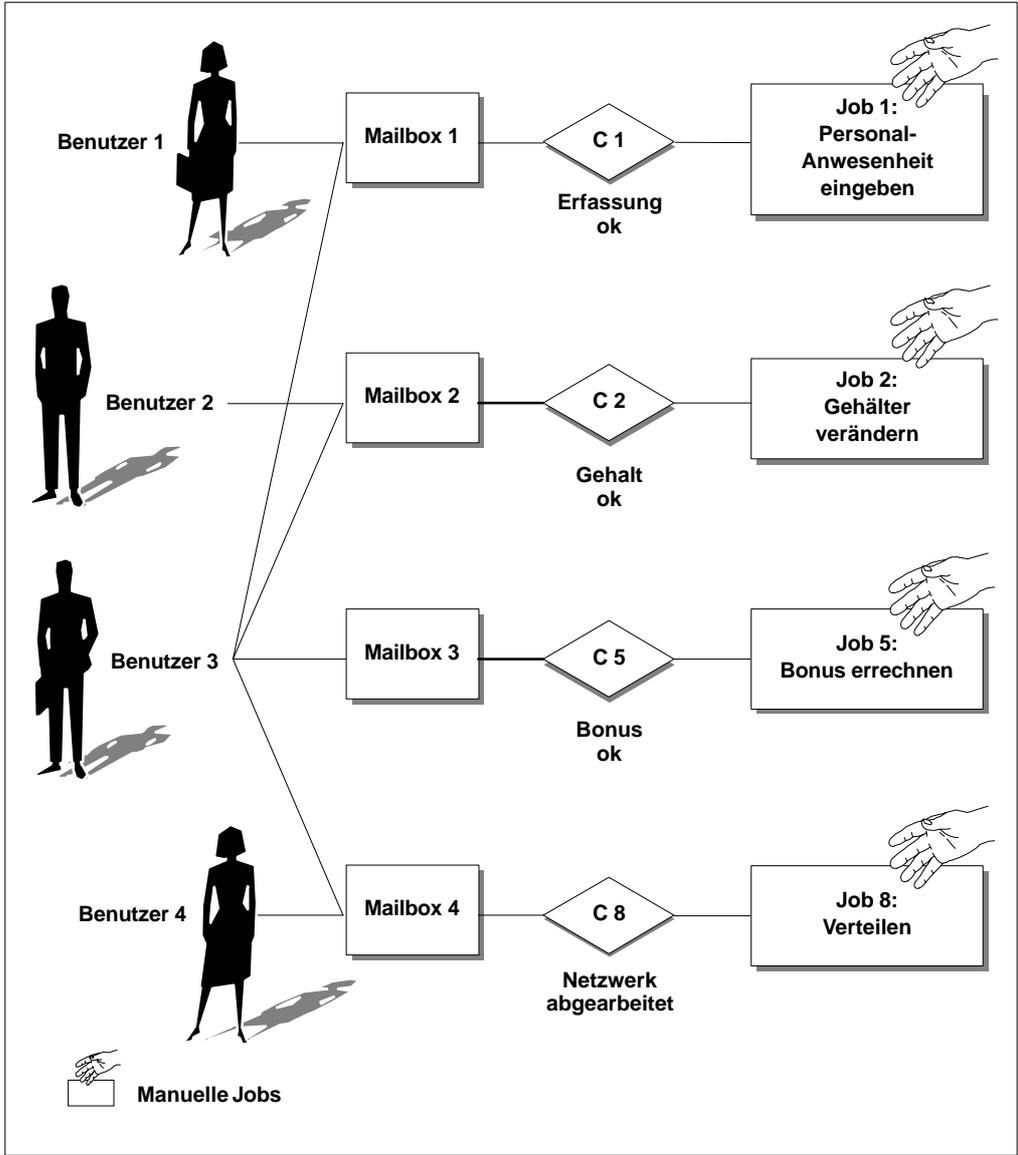


Abbildung 2-8: Das Konzept der Mailboxen

So wird z.B. **Benutzer 1** (die in der Datenerfassung tätig sein könnte) darüber informiert, daß die **Bedingung 1 (Erfassung-OK)** nicht erfüllt ist. Sie kann dann die notwendigen Maßnahmen durchführen, indem sie die Daten für die Personalanwesenheit vervollständigt. Danach quittiert sie das in ihrer Mailbox, und die Verarbeitung wird fortgesetzt. **Benutzer 3** (der der Assistent des Personal-Chefs sein könnte) wird über jede nicht erfüllte Bedingung benachrichtigt und kann daher die Abarbeitung des gesamten Job-Netzwerkes überwachen. **Benutzer 3** und **4** werden benachrichtigt, wenn das Netzwerk vollständig abgearbeitet ist und die Belege verteilt werden können.

Bis zu 10 Mailboxen können mit einer Benutzer-Kennung verknüpft werden.

## Ressourcen

---

Um die Durchführung von Jobs von der Verfügbarkeit von **Ressourcen** abhängig zu machen, können die Funktionen des Entire System Server benutzt werden.

**Ressourcen** müssen zuerst in der Systemverwaltung definiert werden, bevor sie innerhalb von Entire Operations als Vorbedingung von Jobs verwendet werden können. Ressourcen regeln den Verarbeitungsfluß und können in beliebiger Anzahl definiert werden. Sie verhindern, daß Jobs parallel laufen, wenn alle sonstigen Voraussetzungen (z.B. Zeitfenster oder logische Eingabebedingungen) erfüllt sind. Weitere Informationen entnehmen Sie dem Abschnitt **Resources** in der *Administration Documentation*.

Im Beispiel des Netzwerkes aus der Lohnbuchhaltung (Abbildung 2-6 auf Seite 20) kann verhindert werden, daß **Job 6** (Generiere Personal-Liste) und **Job 7** (Generiere Rechnungsbelege) parallel laufen. Hierzu kann eine Ressource definiert werden (z.B. **CPU-Zeit**), die eine in einer Anfangsmenge von 100 Einheiten vorhanden ist und von der jeder der beiden Jobs 60 Einheiten benötigt. Beide Jobs werden dann sequentiell abgearbeitet werden.

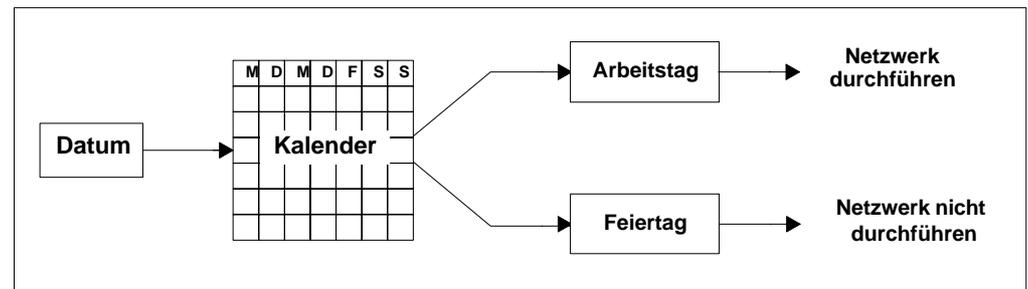
Eine ausführlichere Beschreibung entnehmen Sie dem Kapitel **Ressourcen** im *Entire Operations Benutzerhandbuch*.

## Kalender

Eine beliebige Anzahl von **Kalendern** kann auf sehr komfortable Art und Weise innerhalb des Systems definiert und verwaltet werden. Diese können dann entweder speziellen Eigentümern oder aber dem ganzen System zur Verfügung stehen.

Die Verwendung von Kalendern dient nur einem einzigen Zweck, nämlich zwischen **Arbeitstagen** und **Feiertagen** unterscheiden zu können. Entire Operations führt keine Aktivitäten an Feiertagen durch. Sie können festlegen, was mit einem Job-Netzwerk geschehen soll, das an einem solchen Tag eigentlich eingeplant war (Aktivierung vorher, Aktivierung nachher, ausfallen lassen).

Wenn Sie andererseits ein Job-Netzwerk z.B. jeden Freitag ausführen möchten brauchen Sie hierfür keinen Kalender.



## Zeitpläne

**Zeitpläne** beinhalten die Datumsangaben für die geplante Ausführung von Job-Netzen. Sie können sowohl periodische als auch explizite Datumsangaben umfassen. Eine beliebige Anzahl von Zeitplänen ist definierbar, und ein Zeitplan kann von verschiedenen Job-Netzwerken referenziert werden.

Falls ein Zeitplan auf einem vordefinierten Kalender aufbaut, können Datumsangaben zur Ausführung von Job-Netzwerken sogar relativ zu Feiertagen gemacht werden (z.B. letzter Arbeitstag eines Monats).

## Symbol-Tabellen und Symbole

---

*Symbol-Tabellen* sind benutzerdefinierte Tabellen, die jeweils eine Liste von Symbolnamen mit ihren aktuellen Werten beinhalten. Diese Tabellen werden immer benutzt, wenn die auszuführenden Job-Steueranweisungen zum Aktivierungs- oder Ausführungszeitpunkt erst aufgebaut werden sollen ('dynamische Generierung von Job-Steueranweisungen'). Der Vorteil solcher Symbol-Tabellen ist, daß sie nur einmal gepflegt werden müssen, aber in einer großen Anzahl von Jobs referenziert werden können.

Sie können eine beliebige Anzahl von Symbol-Tabellen definieren und sie einfach dadurch benutzen, indem sie ihren Namen in der Definition der entsprechenden Job-Netzwerke angeben.

Bei der Definition von Symbolen kann spezifiziert werden, daß bei der Aktivierung der sie verwendenden Job-Netzwerke eine Aufforderung an eine Mailbox zur Dateneingabe erfolgen soll: die am Bildschirm eingegebenen Daten werden dann in die auszuführenden Job-Steueranweisungen übernommen.

Bei jeder Netzwerk-Aktivierung wird eine Kopie der verknüpften Symbol-Tabelle(n) erstellt. Damit wird es Ihnen möglich, Netzwerke mit unterschiedlichen Parametersätzen zur Ausführung vorzubereiten, und das sogar über einen längeren Zeitraum im voraus.

Wie können nun Symbole in den Job-Steueranweisungen benutzt werden? Immer wenn ein Symbolname in den Job-Steueranweisungen oder in einer Prozedur vorkommt, wird er durch seinen aktuellen Wert aus der Symbol-Tabelle ersetzt. Den Zeitpunkt dieser Ersetzung können Sie selbst anhand zweier Fluchtsymbole bestimmen: entweder den Zeitpunkt der Netzwerk-Aktivierung oder aber den Zeitpunkt der tatsächlichen Job-Ausführung.

Es gibt in Entire Operations auch eine große Anzahl vordefinierter (eingebauter) Symbole. Weitere Informationen siehe Abschnitt **Vordefinierte Symbole** im Kapitel **Symbole**.

Die Symbole selbst werden in verschiedenen Tabellen (wie in STEPLIBs) gesucht. Symbole können rekursiv andere Symbole beinhalten; Systemvariablen können ebenfalls zur Bildung von Symbolwerten herangezogen werden.

Symbol-Tabellen gehören Eigentümern. Ein Eigentümer kann eine beliebige Anzahl von Symbol-Tabellen besitzen. Ein Benutzer kann alle Symbol-Tabellen aller Eigentümer verändern, für die er autorisiert ist.

Symbole können mit Hilfe sogenannter APIs (Application Programming Interfaces) von beliebigen Natural Programmen abgefragt und modifiziert werden. Wird ein solches Natural Programm als Teil eines Entire Operations Job-Netzwerks ausgeführt, so können aktive Symbol-Tabellen sogar während der laufenden Ausführung von Job-Netzwerken verändert werden.

## Laufnummern

---

Aktive Objekte in Entire Operations werden zusätzlich durch eine Laufnummer identifiziert, die ihnen automatisch bei der Erzeugung eines aktiven Objekts zugewiesen wird. Aktive Netzwerke oder Jobs werden bei einer Aktivierung eines Netzwerks oder eines Jobs erzeugt.

Laufnummern sind auf der Job-Netzwerk-Ebene eindeutig.

Laufnummern können auch geplanten Aktivierungen zugewiesen werden. In der Planungsphase sind für eine gegebene Laufnummer keine aktiven Jobs vorhanden.

- Es gibt keine Garantie dafür, dass die Nummerierung der Netzwerk-Aktivierung mit den Aktivierungszeiten ansteigt.

### Laufnummern-Bereich

Bei einem neuen Netzwerk beginnt die Erzeugung von Laufnummern mit 1.

Wenn die höchstzulässige Laufnummer erreicht ist, beginnt die Nummerierung aktiver Netzwerke wieder mit 1.

Importierte Job-Netzwerke fahren mit der Nummerierung aus der früheren Umgebung fort.

Die höchstzulässige Laufnummer kann in den Systemverwaltungsdiensten definiert werden. Weitere Einzelheiten siehe Abschnitt **Entering the Default Setting Utility** im Kapitel **Entire Operations Defaults** in Ihrer *Entire Operations Administration Documentation*.

### Reservierte Laufnummern

Einige Laufnummern werden von Entire Operations zu speziellen Zwecken verwendet. Diese Nummern können nicht für irgendwelche Benutzer-Netzwerke und Jobs verwendet werden.

Laufnummer	Äquivalent	Verwendung
-1	abs	Interne Darstellung der Bedingungs-Referenz 'absolut'
-2	void	Interne Darstellung der Bedingungs-Referenz 'void' (leer)
-3	K-RUN-MACROTEST	Beim Testen von Makro-Programmen verwendet.
-4	K-RUN-PREGENERATED	Für die Speicherung vorgenerierter JCL verwendet.

## ENTIRE OPERATIONS KOMponentEN

Entire Operations umfaßt folgende Komponenten:

- Die Master-Datenbank;
- Die externen Job-Steueranweisungen
- Die aktive Datenbank;
- Der Monitor;
- Das Betriebssystem (CPU)
- Die Benutzerschnittstelle;
- Die Log-Funktionalität;
- Die Report-Funktionalität;
- Der Editor.

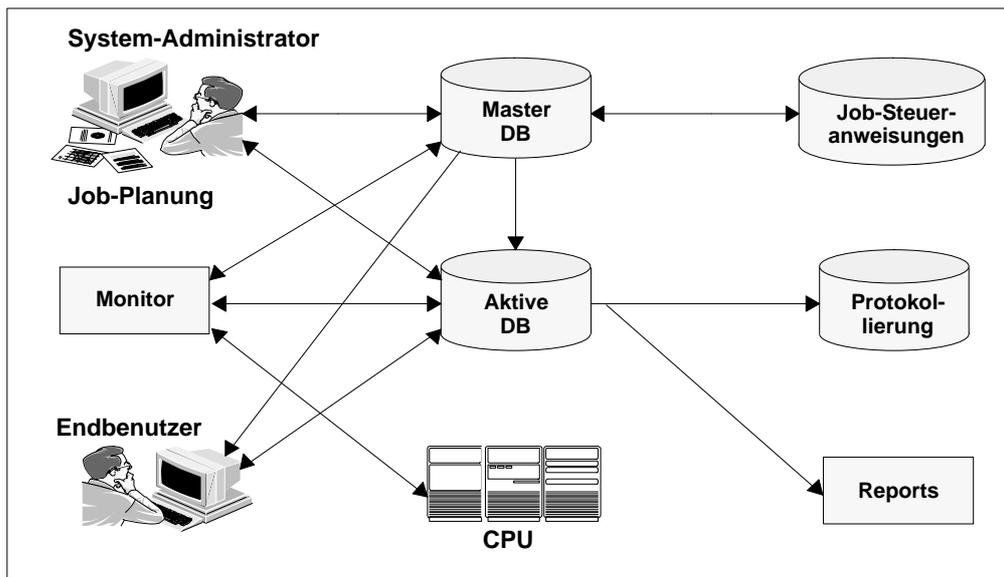


Abbildung 3-1: Entire Operations Komponenten

Jede Komponente soll nun kurz beschrieben werden.

## Die Master-Datenbank

---

Alle Definitionen und Informationen, die Benutzer, Job-Netzwerke, Jobs und Zeitpläne betreffen, werden in der Master-Datenbank abgelegt. Die Master-Datenbank ist eine Adabas Datei, was automatisch Benutzer-Synchronisation, Datenintegrität, Datenkomprimierung, automatische Dateierweiterung und automatische Wiederanlauffähigkeit bedeutet. Alle gespeicherten Objekte können am Bildschirm aus jeder der Host-Ablaufumgebungen Complete, CICS, IMS, TSO, TIAM und UTM sowie unter UNIX verwaltet werden. Zu diesen Objekten zählen:

- Benutzer-Profile;
- Definitionen der Job-Netzwerke;
- Job-Definitionen;
- Eingabe- und Ausgabe-Bedingungen;
- Definitionen der Ressourcen;
- Zeitpläne;
- Kalender;
- Symbol-Tabellen.

## Die Externen Job-Steueranweisungen

---

Mit Entire Operations können Job-Steueranweisungen unverändert in Job-Netzwerke integriert werden, sie können sogar an ihrem ursprünglichen Speicherort bleiben. Hier werden sequentielle Dateien, PDS-, VSE/ESA- und LMS-Bibliotheken sowie die Speicherformen NATURAL und CA-LIBRARIAN unterstützt. Unter dem Betriebssystem UNIX können beliebige Shell-Prozeduren in die Steuerung von Entire Operations einbezogen werden.

Ein Kopieren der Job-Steueranweisungen in die Master-Datenbank ist über die **IMPORT** Funktion möglich, und sollte immer dann benutzt werden, wenn besondere Sicherheitskriterien dies erfordern (Zugriff ist dann nur noch über Natural Security möglich) oder falls die Job-Steueranweisungen mit der Master-Datenbank gesichert werden sollen.

## Die Aktive Datenbank

Immer wenn ein Job-Netzwerk aktiviert wird, wird eine Kopie in die aktive Datenbank eingestellt. Ein Netzwerk wird hierbei entweder automatisch vom Entire Operations Monitor entsprechend der Zeitplan-Daten oder aber durch einen Benutzer auf Anforderung aktiviert. Die aktive Datenbank kann daher auch mehrere Kopien desselben Job-Netzwerks beinhalten, jede von ihnen wird durch eine eindeutige Laufnummer identifiziert.

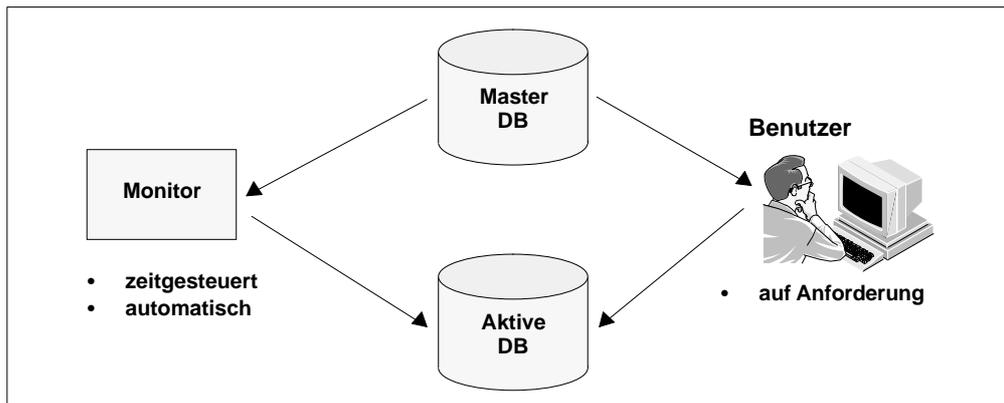


Abbildung 3-2: Die Aktive Datenbank

Folgende Informationen werden abgelegt:

- aktuelle Definition eingeplanter Job-Netzwerke zusammen mit ihren aktuellen Symbol-Tabellen;
- aktive Job-Steueranweisungen (dies bedeutet, daß alle Informationen über Job-Steueranweisungen aus externen Speichermedien wie z.B. PDS, LMS, VSE-LIBRARIAN oder UNIX-Dateien in die aktive Datenbank kopiert werden);
- aktueller Status der Eingabe- und Ausgabe-Bedingungen;
- aktueller Job-Status.

Sie können auf die aktive Datenbank auf die gleiche Art und Weise zugreifen und ihre Informationen modifizieren wie auf die Master-Datenbank. Änderungen an einem Objekt der aktiven Datenbank sind nur für den zugehörigen Netzwerklauf wirksam und haben keinen Einfluß auf die Definitionen der Netzwerke und Jobs in der Master-Datenbank. Hiermit wird es Ihnen möglich, Änderungen vorzunehmen, die nur an ganz bestimmten Produktionstagen Gültigkeit haben sollen.

## Der Monitor

---

Der Monitor ist das Herz von Entire Operations. Er ist ein komplexes Programm, das periodisch aktiviert wird und die Definitionen der Master-Datenbank daraufhin überprüft, ob eine Verarbeitung ansteht. Der Monitor aktiviert Netzwerke und bringt Jobs unter Berücksichtigung ihrer Vorbedingungen zur Ausführung; er kontrolliert die Abarbeitung von Job-Netzwerken, auch dann, wenn sie auf verschiedenen Rechnern stattfindet.

Der Monitor führt die folgenden Funktionen durch:

- er aktiviert automatisch eingeplante Netzwerke (und kopiert sie dabei in die aktive Datenbank);
- er überprüft Zeitfenster für die Ausführung von Jobs und Job-Netzwerken;
- er überprüft Eingabebedingungen und Ressourcen;
- er bringt Jobs unter Berücksichtigung ihrer (internen) Prioritäten zur Ausführung;
- er überwacht Jobs in den verschiedenen Warteschlangen des Betriebssystems;
- er analysiert den Jobende-Status, stellt das Eintreten von Ereignissen fest und stößt entsprechende Systemaktionen (Bedingungen setzen, Meldungen schicken, Programme starten) an;
- er protokolliert alle wichtigen Informationen.

Es ist möglich, die Funktionen des Entire Operations Monitors auf mehrere Unterfunktionen (Subtasks) zu verteilen. Durch dieses Subtasking können Verarbeitungsprozesse parallelisiert und so Multiprozessor-Architekturen zur Performanceverbesserung ausgenutzt werden. Die Verteilung der typischen Monitorfunktionen nimmt der Systemadministrator vor.

## Das Betriebssystem

---

CPU-basierte Verarbeitung wird vom Entire Operations Monitor auf den zugrundeliegenden Betriebssystemen initiiert: Jobs, Tasks, Prozeduren oder Natural-Programme werden gestartet. Der Monitor schickt hierzu entsprechende Anforderungen an den ausführenden Entire System Server Knoten, eventuell über Entire Net-Work auch netzwerkweit.

## Die Benutzerschnittstelle

Entire Operations besitzt eine menü-gesteuerte, leicht zu bedienende Benutzerschnittstelle, welche jedem Benutzer des Systems alle notwendigen Funktionen zur Verfügung stellt, für die er aufgrund seines Authorisierungsprofils berechtigt ist. Großer Wert wurde darauf gelegt, die Arbeit mit dem System so komfortabel wie irgendetmöglich zu machen, insbesondere für Endbenutzer.

Die folgende Abbildung zeigt einen typischen Entire Operations Bildschirm — den Bildschirm *Job-Verwaltung* innerhalb der Funktion *Netzwerk- und Job-Definition*:

The screenshot shows a terminal window with the following content:

```

02.04.02          *** Entire Operations ***          15:14:31
Eigentuerer EXAMPLE          Jobverwaltung          Netzwerk B60-FLOW
-----
Bef B R PU Job          Typ Beschreibung          Datei oder Bibliothek Member
*-----
-          JOB-01          MAC Wo alles anfaengt .   EOR-T210          B60-M02
-   B1 P   JOB-012          MAC Abhängig von JOB-01 . EOR-T210          E57-M01
-   B1     JOB-013          MAC Abhängig von JOB-012 EOR-T210          E57-M01
-   B1     JOB-014          MAC Abhängig von JOB-013 EOR-T210          E57-M01
-   B1     JOB-015          DUM Abhängig von JOB-014
-   B1     JOB-019          MAC Abhängig von JOB-01   EOR-T210          E57-M01
-   B2     JOB-02          MAC Dep. JOB-15, JOB-19   EOR-T210          E57-M02
-   B1     JOB-03          NAT Abhängig von JOB-02   EOR-T210          E57-P01
-   B1     JOB-04          MAC Abhängig von JOB-03   EOR-T210          E57-M01
-   B1     JOB-05          DUM Abhängig von JOB-04
-   B1     JOB-06          MAC Wo alles endet        EOR-T210          E57-M02
-
***** Ende der Daten *****
A Abhaeng. C Kop. D Loe. E Edit G JCL vorgeh. I Eing.Beding. L Ress. M Aendern
O Jobende-Prf + Akt P Beschr. R Aktivieren S Zeitpl.Parameter U Zusaeztl. Log
Befehl =>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      Help  Add   End           Save      Up      Down           Menu
  
```

Callouts in the image point to the following elements:

- Kopfzeilen**: Points to the header information (date, time, title).
- PF-Tasten**: Points to the function key row (PF1-PF12).
- Zeilenbefehle**: Points to the command line (Befehl =>).
- Befehlszeile**: Points to the command input area.
- Datenzeilen**: Points to the data table rows.

Abbildung 3-3: Entire Operations Benutzerschnittstelle

Die folgenden Merkmale sind realisiert:

## Einheitlicher Maskenaufbau

Alle Bildschirmmasken in Entire Operations haben die gleiche Struktur und umfassen folgende Teile:

- **Kopfzeilen** zur Identifizierung des Benutzers und der aktuellen Funktion;
- **Datenzeilen** mit vorangestelltem Zeilenbefehlsfeld;
- Zeilen mit der Aufzählung aller möglichen **Zeilenbefehle**;
- eine **Befehlszeile** zur Eingabe von Direktbefehlen;
- eine Zeile mit der Auflistung der gerade unterstützten **PF-Tasten** sowie deren Beschreibung.

## Zeilenbefehle

Sie werden in das erste Feld einer Datenzeile in einer Entire Operations Bildschirmmaske eingegeben und erlauben Anwendern, bestimmte Funktionen auf dem so ausgewählten Objekt durchzuführen. Jede Bildschirmmaske enthält eine Auflistung der lokal verfügbaren Zeilenbefehle.

## Direktbefehle

Sie werden in eine dafür vorgesehene Zeile im unteren Teil jeder Bildschirmmaske eingegeben und erlauben es dem Anwender, die Menüführung des Systems zu umgehen. Auf jedes Menü innerhalb Entire Operations kann man so direkt zugreifen. Bei der Ausführung haben Direktbefehle eine höhere Priorität als Zeilenbefehle und PF-Tasten-Funktionen.

## PF-Tasten

Verschiedene Funktionen zur Verwaltung, Navigation und zum Blättern sind über PF-Tasten verfügbar. Auf jedem Entire Operations Bildschirm sind die verfügbaren PF-Tasten zusammen mit der Beschreibung ihrer Funktion aufgelistet.

## Fenster-Technik

```

02.04.02          *** Entire Operations ***          15:26:33
Eigentuerer EXAMPLE      Eingabebedingungen          Job JOB-02
Netzwer
-----
B Bedin          EOR0920 - Objekt ist schon definiert          +
                   Definition Master-Eingabebedingung          ! n.Rtn
  E60-J
  E60-J          Eigentuerer ==> EXAMPLE_____          !
  BRY-1          Netzwerk ==> E60-FLOW_____          !
                   Bedingung ==> BRY-2_____          Lauf ==> _____          !
                   Referenz ==> _____          !
  Verwend
  Muss E          Eingabebedingung: Benutzer-Routine          ! : _          !
  Abhaeng          Bedingung ==> BRY-2          !          !
  Benut           Lauf ==>          ! > _          ! *****
  D Loesc          Datei          ! > _          !
  Ben.-           Wert wird bestimmt durch          ! > _          !
  Job-V           NATURAL Bibliothek ==> _____          !          !
  Befehl          Benutzer-Routine ==> _____          !          !
  -PF1---P
Enter-PF         Help A          PF1 Help PF3 End          PF9 Delete          ! 2--          !
He              + enu

```

Abbildung 3-4: Fenster-Technik

Entire Operations verwendet das Konzept von 'Bildschirm-Fenstern', um es Anwendern des Systems auf möglichst einfache Weise zu gestatten, Daten einzugeben und Objekte aus Listen auswählen zu können.

## Die Online-Hilfe

Entire Operations stellt eine umfassende Online-Hilfe zur Verfügung, die es einem Anwender des Systems ermöglicht, entweder auf *Feldebene* oder für *ganze Bildschirmmasken* Erklärungen zu erhalten.

Die Hierarchie der Hilfebildschirme reflektiert die Hierarchie der Entire Operations Systembildschirme. Dies bedeutet, daß Sie die Online-Hilfe von einem beliebigen Systembildschirm aufrufen können, worauf Sie den für diesen Bildschirm definierten Hilfetext sofort angezeigt bekommen. Dann können Sie zum Systembildschirm zurückkehren oder weitere Hilfe anfordern.

Als Alternative dazu können Sie die Online-Hilfe am oberen Ende der Hierarchie aufrufen, indem Sie die Option *Hilfe* im *Hauptmenü* auswählen. Damit wird das Inhaltsverzeichnis der Hilfe mit einer Liste der auswählbaren Objekte und eine kurze Erläuterung ihrer Bedeutung angezeigt:

```

02.04.02                               Entire Operations HILFE                               15:32:05
----- Inhaltsverzeichnis -----
Option ==>                               H00000

          E N T I R E   O P E R A T I O N S   H I L F E

Waehlen Sie bitte eines der folgenden Themen, um ausfuehrlichere
Informationen zu erhalten:
 1  Hilfe zur Hilfe      - Das Entire Operations Hilfesystem
 2  Die Masken          - Funktionalitaet der Masken
 3  Netzwerke und Jobs  - Definition von Netzwerken und Jobs
 4  Kalender            - Definition und Verwendung der Kalender
 5  Aktive Jobs         - Anzeige und Veraenderung der aktiven Jobs
 6  Symboltabellen     - Editieren der Symboltabellen
 7  Logging             - Auswahl und Anzeige von Log-Informationen
 8  Dynamische JCL     - Die Verwendung von dynamischer JCL
 9  System              - Systemverwaltungsdienste
10  Aktive Bedingungen - Verwaltung der aktiven Bedingungen
11  Editor              - Der Entire Operations Editor
12  Berichte            - Entire Operations Berichtsfunktionen
13  Direktbefehle      - Die Verwendung von Direktbefehlen
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
          Quit  End

```

Abbildung 3-5: Entire Operations Online-Hilfe

## **Deutsch und Englisch**

Entire Operations ist in deutscher und englischer Sprache verfügbar: dies schließt das gesamte Online-System mit allen Masken, Hilfe-Informationen und Fehlermeldungen ein.

## **Farbunterstützung**

Farben werden intensiv zur Hervorhebung wichtiger Informationen, Warnungen usw. gegenüber anderen Informationen verwendet.

## Die Log-Funktionalität

---

Entire Operations bietet eine Log-Funktionalität an, mit deren Hilfe jedes Ereignis während einer Netzwerkdurchführung aufgezeichnet werden kann. Es protokolliert darüberhinaus alle manuellen Veränderungen an den aktiven Jobs, wie z.B. Änderungen an Job-Steueranweisungen oder Modifikationen an Symbol-Tabellen. Diese Informationen sind dann online verfügbar und können für Berichte und statistische Zwecke genutzt werden.

Log-Informationen können nach verschiedenen Selektionskriterien ausgewertet werden: nach Job-Netzwerken und Jobs, nach Datums- und Zeitbereichen und nach Benutzern. Sie können auch in eine sequentielle Datei entladen werden, um von dort aus nach völlig frei wählbaren Kriterien mit beliebigen Werkzeugen weiterverarbeitet zu werden.

Die Aufzeichnung der Job-Steueranweisungen selbst, des Job-Protokolls und/oder mit dem Job verknüpfter Meldungen kann entweder für alle oder aber ausgewählte Jobs angefordert werden.

## Die Report-Funktionalität

---

Die Report-Funktionalität stellt eine große Palette von Berichten zur Verfügung, die insbesondere für die Verfolgung aller durchgeführten Aktivitäten sowie als Grundlage für Zeitplanungen oder sogar für zukünftige Produktionszyklen nützlich sein können.

Berichte bauen auf vorhandenen Log-Informationen auf. Sie können für alle Jobs oder aber für diejenigen, die innerhalb eines Datums-/Zeitfensters ausgeführt wurden angefordert werden, wobei diese Auswahl auf alle beendeten oder aber alle abgebrochenen Jobs eingeschränkt werden kann.

Beschreibungen zu Job-Netzwerken stehen als kurze Übersicht oder aber in ausführlicher Form zur Verfügung.

Die Liste aller Jobs für ein bestimmtes Datum kann angezeigt werden. Dies macht die Vorhersage eines beliebigen zukünftigen Produktionstages möglich.

Alle Berichte können entweder online angezeigt oder aber zu Archivierungszwecken ausgedruckt werden.

## Der Editor

Entire Operations beinhaltet die hierfür angepaßte Version des Software AG Editors, mit dem folgende Objekte erstellt, angezeigt oder verändert werden können:

- Job-Steueranweisungen in der Master-Datenbank, entweder in einer externen Datenhaltung oder aber in einer Natural Datei;
- Job-Steueranweisungen in der aktiven Datenbank;
- Natural Programme und Benutzer-Routinen;
- Online Dokumentation von Netzwerken, Jobs und beliebigen Ereignissen (z.B. als Notizzettel-Information);
- Job-Protokolle und Job-Ausgabelisten (nur im Anzeigemodus).

```

Beschreibung editieren Jb: ABC3A Nw: EORACCT----- Columns 001 072
====>                                     BLAETTERN==> CSR
***** ***** top of data *****
00001 Job 'ABC3B'
00002 -----
00003 Dieser Job erzeugt ein definiertes Ereignis beim Job-Ende und fuehrt
00004 das Programm 'EORCONTI' aus, das in der Installations-Bibliothek
00005 'EORnnn.LOAD' resident ist. Die nötige JCL, die 'EORE*' heisst, ist in
00006 der Installations-Bibliothek 'EORnnn.SRCE' gespeichert.
***** ***** bottom of data *****

```

```

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      End      Rfind Rchan Up      Down Impo Left Right Curso

```

Abbildung 3-6: Entire Operations Editor (Job-Beschreibung)

Der Entire Operations Editor erlaubt die gleiche Behandlung all dieser Objekttypen auf jedem der unterstützten Betriebssysteme und in jeder Ablaufumgebung.



## ENTIRE OPERATIONS FUNKTIONEN

Entire Operations bietet eine große Anzahl menü-unterstützter Funktionen an, die dem Anwender die Definition von Objekten innerhalb des Systems und die Kontrolle und Überwachung der Durchführung von Netzwerken erlauben. Dieses Kapitel gibt eine kurze Einführung in die folgenden Funktionen:

- Das Hauptmenü
- Verwaltung der Job-Netzwerke
- Zeitplanung für Job-Netzwerke
- Job-Verwaltung
- Job-Zeitplanung
- Kalender-Definition
- Verwaltung logischer Bedingungen
- Jobende-Prüfung
- Jobende-Aktionen
- Benutzung von Ressourcen
- Dynamische Generierung von Job-Steueranweisungen
- System-Objekte editieren
- Berichte generieren

## Das Hauptmenü

Nach Anmeldung an Entire Operations sieht der Anwender das *Hauptmenü*. Sobald er hier eine Auswahl trifft, wird er entweder nachgeordnete Funktionsmenüs oder aber eine Auflistung von Objekten angezeigt bekommen, die dann mit Zeilenbefehlen oder aber per PF-Tasten verarbeitet werden können.

```

13.04.02                *** Entire Operations ***                12:07:05
Eigentuemmer EXAMPLE          Hauptmenu                Benutzer-ID BRY
-----
      Hauptmenu                                RZ-Loesungen

1  Netzwerk- und Job-Definition                20  Entire Event Management (V132)
2  Aktive Job-Netzwerke                        21  Entire Output Management (V134)
3  Kalender
4  Log-Information
5  Symboltabellen
6  Systemverwaltung
7  Berichte
8  Import/Export
9  Hilfe

                                           Anwendungen
                                           30  sysmain

Befehl => _____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help           End                                Owner Mail

```

Abbildung 4-1: Hauptmenü

Die Gestaltung des *Hauptmenüs* kann von Benutzer zu Benutzer unterschiedlich sein, je nach Autorisierung in seinem Anwenderprofil.

Eigene Anwendungen können in das *Hauptmenü* integriert werden, was umständliche An- und Abmeldeprozeduren überflüssig macht. Im obigen Beispiel ist *Entire Output Management* Option 21.



## Zeitplanung für Job-Netzwerke

Zeitplanung für ein Job-Netzwerk bedeutet, Datums- und Zeitangaben festzulegen, an denen dieses Netzwerk durchgeführt werden soll. Ein Job-Netzwerk wird aktiviert (d.h. in die aktive Datenbank kopiert) entsprechend seiner eingeplanten Startzeit und unter Berücksichtigung weiterer Plan-Parameter des Netzwerks.

```

03.04.02                *** Entire Operations ***                09:36:32
Eigentuermer EXAMPLE    Definition des Zeitplans
Netzwerk DEMO-NET
-----

Typ                    ==> W      M monatlich W woeentlich E spezielle Tage

Kalender                ==> EXAMPLECAL

frueh. Start           ==> 19:00:00      Zeitplan gueltig ab => _____ 00:00:00
spaed. Start           ==> 21:00:00    ___ Tage spaeter
Endezeit               ==> 22:00:00    ___ Tage spaeter

Anzahl Aktivierungen ==> _____

Aktivieren alle        ==> _____ Minuten
oder aktivieren um ==> _____
                    _____

Befehl => _____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
Help      End      Save      Cal      Menu

```

Abbildung 4-3: Netzwerk-Zeitplanung

Zeitangaben können unter Bezugnahme eines vordefinierten Kalenders und/oder als Liste von expliziten Datumsangaben spezifiziert werden (z.B. 01.02.96, 07.03.96 usw.). Daneben ist die Verknüpfung von Monats- und Tagesangaben ebenfalls möglich: wenn z.B. als Monatsliste '02, 03' und als Tagesliste '01,15' angegeben ist, wird dieses Netzwerk am ersten und fünfzehnten Tag im Februar und März ausgeführt. Feiertage können unter Bezugnahme auf Kalender-Definitionen berücksichtigt werden.

Während der Durchführung eines Job-Netzwerks überprüft Entire Operations, ob alle Vorbedingungen für einen Job erfüllt sind: Zeitfenster, Ressourcen und Eingabebedingungen. Entsprechend dieser Überprüfung wird der Job automatisch gestartet und seine Ausführung überwacht.

## Job-Verwaltung

Innerhalb der Job-Netzwerk-Verwaltung kann der Anwender die Liste aller zu einem Netzwerk gehörenden Jobs anfordern. Er kann dann Jobs hinzufügen, löschen oder aber verändern. Ein Job wird hierbei definiert durch einen (logischen) Namen, einen Typ und seinen Speicherort (PDS-Datei, Natural-Bibliothek usw.) Eine kurze Erläuterung des Jobs kann hinzugefügt werden. Daneben kann für die Ausführung dieses Jobs eine Knoten-Nummer angegeben werden, die dann den Standardwert überschreibt, der bei der Netzwerk-Definition angegeben wurde.

```

03.04.02                *** Entire Operations ***                10:02:22
Eigentuerer EXAMPLE          Jobverwaltung                        Netzwerk E60-FLOW
+ --
!
!
! 2
! 1
! 1
! 1
!
! 1
! 2
! 1
! 1
! 1
!
! 2
!
! 2
!
! **
! rn
! g
+
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
Help Add End Edit Save Spec Symb Impo Copy Menu

```

Abbildung 4-4: Job-Verwaltung

Wenn im Job die Möglichkeit der dynamischen Generierung von Job-Steueranweisungen genutzt werden soll, muß die zu verwendende Symbol-Tabelle ebenfalls angegeben werden (siehe hierzu auch den Abschnitt **Dynamische Generierung von Job-Steueranweisungen** auf Seite 57). Der Anwender kann dann alle derart referenzierten Job-Steueranweisungen einfach über eine zugeordnete PF-Taste editieren.

## Job-Zeitplanung

Die Aktivierung des einzelnen Jobs hängt von seinen Zeitplanungsparametern ab. Genauso wie für ganze Netzwerke können auch für einzelne Jobs (früheste) Startzeiten vergeben werden.

```

03.04.02          *** Entire Operations ***          11:23:00
Eigentuemmer EXAMPLE          Jobverwaltung          Netzwerk E60-FLOW

          Zeitplan-Parameter
Job JOB-012

Geschaetzte Laufzeit ==> _____ Min.   Durchschnitt ==>   1.91
  Letzte Laufzeit   ==>   2.00   2.01   0.03   2.01
                   2.00   2.00   2.00   2.00

Frueheste Startzeit ==> 00:00:00
Spaeteste Startzeit ==> 00:00:00   ___ Tage spaeter
Endezeit            ==> 00:00:00   ___ Tage spaeter

Zeitplanabhaengigkeit ==> N

---PF1---PF3-----PF5-----PF9---PF10---PF11---PF12---
  Help  End          Save          LMsg  Acct   ScDep  Menu

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      Help  Add   End          Save          Up   Down          Menu
  
```

Abbildung 4-5: Job-Zeitplanung

Daneben kann der Anwender angeben, ob ein eingeplanter Job, der aus irgendeinem Grund nicht durchgeführt wurde (z.B. wegen Hardware-Problemen) nochmals eingeplant werden soll: der Job würde dann zum nächsten Planungsdatum zweimal ausgeführt werden. Es kann darüberhinaus bestimmt werden, wie lange Jobs in der aktiven Datenbank auf ihre Ausführung warten dürfen. Auch kann der Anwender dem Job eine geschätzte Laufzeit zuordnen: der Entire Operations Monitor benutzt diese dann zur Berechnung der voraussichtlichen Ende-Zeit des Jobs.

Mit Hilfe der Zeitplan-Angaben eines Jobs ist es Entire Operations möglich, ausgewählte Benutzer des Systems davon zu unterrichten, daß der Start eines Jobs nicht erfolgt ist, um eine vorher definierte späteste Ende-Zeit ('deadline') nicht zu überschreiten.

## Kalender-Definition

Kalender werden in Zeitplänen referenziert, welche ihrerseits in der Netzwerk-Verwaltung definiert sind. Die Anzahl von Kalendern innerhalb des Systems ist beliebig. Kalender können einem Eigentümer gehören oder aber innerhalb des gesamten Systems zur Verfügung stehen. In der Kalender-Verwaltung kann der Benutzer diese hinzufügen, löschen oder verändern (systemweite Kalender können nur vom Administrator verändert werden).

```

22.03.02                *** Entire Operations 4.1.1 ***                15:32:11
Eigentmr SN            Kalender DEMOCAL    Jahr 2002

          Januar                Februar                Maerz
Montag          7 14 21 28          4 11 18 25          4 11 18 25
Dienstag       1  8 15 22 29          5 12 19 26          5 12 19 26
Mittwoch       2  9 16 23 30          6 13 20 27          6 13 20 27
Donnerstag     3 10 17 24 31          7 14 21 28          7 14 21 28
Freitag        4 11 18 25          1  8 15 22          1  8 15 22 29
Samstag        5 12 19 26          2  9 16 23          2  9 16 23 30
Sonntag        6 13 20 27          3 10 17 24          3 10 17 24 31

          April                Mai                Juni
Montag         1  8 15 22 29          6 13 20 27          3 10 17 24
Dienstag       2  9 16 23 30          7 14 21 28          4 11 18 25
Mittwoch       3 10 17 24          1  8 15 22 29          5 12 19 26
Donnerstag     4 11 18 25          2  9 16 23 30          6 13 20 27
Freitag        5 12 19 26          3 10 17 24 31          7 14 21 28
Samstag        6 13 20 27          4 11 18 25          1  8 15 22 29
Sonntag        7 14 21 28          5 12 19 26          2  9 16 23 30

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help           End           Up           Down

```

Abbildung 4-6: Kalender-Definition

Ein Kalender wird durch seinen Namen und eine Jahresangabe definiert. Die Definition neuer oder die Verwaltung bestehender Kalender besteht dann einfach darin, Feiertage zu markieren. Im obigen Beispiel sind alle Feiertage hell unterlegt.

Entire Operations berücksichtigt diese Feiertage dann dadurch, daß es Netzwerke, die für diese Tage eingeplant wurden, nicht aktiviert. Weitergehende Regelungen für diesen Fall können getroffen werden, wie z.B. 'aktiviere Netzwerk am folgenden Werktag'.

## Verwaltung logischer Bedingungen

---

Logische Bedingungen stellen Abhängigkeiten zwischen Jobs dar. Sie werden mit einer dafür vorgesehenen Verwaltungsfunktion definiert, gelöscht oder aber verändert. Mit einem Job kann eine beliebige Anzahl logischer Bedingungen verknüpft werden. Eine logische Bedingung kann zwei Stati annehmen, die dann die weitere Verarbeitung innerhalb von Entire Operations bestimmen: WAHR (Bedingung existiert) oder FALSCH (Bedingung existiert nicht).

Jede Bedingung wird durch einen Namen und ein Referenzdatum identifiziert. Dies erlaubt es dem Entire Operations Monitor, zwischen gleichen Ereignissen zu unterscheiden, die zu verschiedenen Zeitpunkten eingetreten sind. Zeitangaben können entweder in relativen oder absoluten Einheiten gemacht werden. Alle relativen Angaben werden in absolute umgerechnet, wenn der Job in die aktive Datenbank eingestellt wird. Logische Bedingungen können auf zwei Arten verwendet werden:

- Als Eingabe-Bedingungen;
- Als Ausgabe-Bedingungen.

## Eingabe-Bedingungen

Sämtliche Eingabe-Bedingungen müssen erfüllt sein, bevor Entire Operations einen zuvor aktivierten Job zur Ausführung bringen kann. Um nun zwei Jobs zu verknüpfen, muß eine Eingabe-Bedingung des einen Jobs mit als Ausgabe-Bedingung seines Vorgängers definiert werden. Eine Eingabe-Bedingung kann durch ein CPU-basiertes Ereignis oder durch einen manuellen Benutzereingriff erfüllt werden.

```

03.04.02                *** Entire Operations ***                12:39:56
Eigentuerer EXAMPLE    Eingabebedingungen                        Job E05-J02-UR
Netzwer
-----
B Bedin                Master-Eingabebedingung aendern                n.Rtn
m E05-U
_ EX-1   Eigentuerer ==> EXAMPLE
_ EX-2   Netzwerk    ==> E05-IOC-04
_ EX-3   Bedingung   ==> E05-UR01-NOK_____ Lauf ==> _____
_        Referenz    ==> RUN_____
_
_        Verwendung (markiere mit Y oder N)
_        Muss Exist.: Y   Exklusiv: _   Loeschen n. Verwendung: _
_
_        Abhaengig von
*****
D Loesc   Benutzer-Routine ==> _           Mehrfache Suffixe ==> _           *****
         Dateiexistenz   ==> _           Mailbox                ==> _
         Ben.-Sch. (BS2000) ==> _         Symbol-Wert            ==> _
         Job-Var. (BS2000) ==> _
Befehl
Enter-PF  -PF1---PF2---PF3-----PF5-----PF9---PF10---PF12--
He        Help Add End      Save          Xref ScDep Menu          F12---
                                         enu

```

Abbildung 4-7: Eingabe-Bedingungen

Neben dem Namen und dem Referenzdatum kann der Anwender auch eine Mailbox mit einer Bedingung verknüpfen. Entire Operations wird dann automatisch alle Benutzer dieser Mailbox über alle ihre zugeordneten, ausstehenden Bedingungen informieren.

Jedem Benutzer des Systems können bis zu 10 Mailboxen zugeordnet werden: er wird so die Liste aller Meldungen sehen können, die an eine dieser Mailboxen geschickt worden ist

Daneben kann der Benutzer festlegen, in welchem Status eine Bedingung sein muß, damit ein Job zur Ausführung gebracht werden kann (WAHR oder FALSCH), ob ein Job warten soll, bis ihm diese Bedingung exklusiv zur Verfügung steht (dies kann benutzt werden, um die parallele Ausführung von Jobs zu verhindern, die die gleiche Bedingung benutzen) und ob Entire Operations am Jobende die Bedingung automatisch zurücksetzen soll.

Vor der Durchführung eines Jobs überprüft der Entire Operations Monitor automatisch alle Eingabebedingungen. Daneben können sie diese Prüfung auch von einer Natural-Benutzeroutine durchführen lassen: diese muß dazu bei der Definition der logischen Bedingung angegeben werden.

## Ausgabe-Bedingungen

Ausgabe-Bedingungen werden vom Entire Operations Monitor automatisch gepflegt, wenn das ihnen zugeordnete Ereignis eingetreten ist. In diesem Fall werden alle Jobs gestartet, für die diese Bedingung einzige Eingabe-Bedingung ist. Ereignisse und Ausgabe-Bedingungen werden in Entire Operations innerhalb der Jobende-Prüfungen festgelegt (siehe den Abschnitt **Jobende-Prüfung**, unten).

```

03.04.02                *** Entire Operations ***                13:05:56
Eigentuemmer EXAMPLE    Jobende-Pruefung + Aktionen            MVS/ESA    Job JOB-01
Netzwerk    E60-FLOW                                Lauf        Datum
-----
B Aktion                                                    t AA
-
-
-      P
-      Ausgabebedingungen
-      Bef Status Name der Bedingung Referenz Lauf
-      _ Setze E60-JOB1-0 RUN
-      _ Setze E60-JOB2-02 RUN
-      _ Loe. E60-JO15-0 RUN
-      _ Loe. E60-JO19-0 RUN
-      _ Loe. E60-JOB1-0 RUN
*****
c C P
- PRS
-----
A Aktivierung
P Beschreib.
-
-      D Loeschen M Aendern W Wo benutzt
-      Befehl => Enter-PF1---PF2---PF3---PF5-----PF7---PF8----PF12-
Enter-PF1---PF Help Add End Save Up Down Menu 2---
Help Ad u
  
```

Abbildung 4-8: Ausgabe-Bedingungen

Genau wie im Fall der Eingabe-Bedingungen werden auch Ausgabe-Bedingungen durch Namen und Referenzdatum definiert. Darüberhinaus kann der Benutzer bestimmen, ob die Ausgabe-Bedingung bei Eintreffen des zugeordneten Ereignisses (auf den Wert WAHR) gesetzt oder (auf den Wert FALSCH) zurückgesetzt werden soll.

Bis zu 20 Ausgabe-Bedingungen können mit einem einzelnen Ereignis verknüpft werden.

## Jobende-Prüfung

Die Jobende-Prüfung bezeichnet die Verarbeitung, mit der Entire Operations das Jobende feststellt.

```

03.04.02                *** Entire Operations ***                13:56:21
Eigentuermer EXAMPLE    Jobende-Pruefung + Aktionen    MVS/ESA    Job JOB-02
Netzwerk    E60-FLOW                                Lauf        Datum
-----
B Aktion      Step      wird geprueft auf                                bedeutet AA
_   P         STEP1    Condition Code = C0000                            ok
_                                     Auftreten von String Bibliothek voll            n.ok
_
_
_
_
_
_
_
***** Ende der Daten *****
_   C P        Alle Pruefungen ok
_   PRS       Mindestens eine Pruefung nicht ok
-----
A Aktivierung C Beding. D Loesch  E Edit.Ben.Rtn  J Job Var.  M Aendern O And.
P Beschreib. R Fehlerbeh. S Sysout T Output Mgmt. U Ben.-Nachr. X Aktion-Rtn

Befehl => _____
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      Help  Add   End           Save           Up           Down           Menu

```

Abbildung 4-9: Jobende-Prüfung und Aktionen

Unmittelbar nach Jobende untersucht Entire Operations, ob benutzer-definierte Ereignisse eingetreten sind. Solche Ereignisse können sein:

- ein Return-Code wurde in einem bestimmten Arbeitsschritt innerhalb eines Jobs gesetzt;
- ein Return-Code wurde in einem beliebigen Arbeitsschritt eines Jobs gesetzt ;
- eine Zeichenkette wurde im Protokoll oder in den Ausgabe-Listen des Jobs gefunden;
- eine Natural-Benutzerroutine wurde ausgeführt, welche ihrerseits den Jobende-Status durch das Setzen eines Return-Codes anzeigt. Diese Benutzerroutine kann:
  - das Job-Protokoll oder die Ausgabelisten selbst untersuchen,
  - alle vom Job erzeugten Daten lesen,
  - Systemfunktionen durchführen,
  - Meldungen verschicken.

## **Standard-Prüfungen**

Je nach Betriebssystem, auf dem der Job durchgeführt wurde, unternimmt Entire Operations eine Reihe von Standard-Prüfungen zur Feststellung des Job-Ergebnisses. Für OS/390-Systeme werden z.B. Systemabbrüche oder Syntaxfehler in den Job-Steueranweisungen automatisch erkannt. Diese Standard-Prüfungen werden für jeden Job durchgeführt, unabhängig davon, ob daneben noch benutzer-definierte Prüfungen für diesen Job angefordert sind.

## Jobende-Aktionen

---

Für jedes definierte Ereignis kann der Benutzer festlegen, wie Entire Operations darauf reagieren soll. Folgende Möglichkeiten stehen zur Verfügung:

- verknüpfte Ausgabe-Bedingungen automatisch setzen oder zurücksetzen (siehe den Abschnitt **Ausgabe-Bedingungen** auf Seite 52;
- eine Meldung schicken an einen bestimmten Betriebssystem-Benutzer, die System-Konsole, sowie eine Entire Operations Mailbox, an einen Benutzer des Bürokommunikationssystems der Software AG Con-nect oder an eine E-Mail-Adresse.
- Job-Protokoll und Ausgabelisten löschen oder drucken;
- Fehlerbehebungsmaßnahmen einleiten (im Falle eines Job-Abbruchs).
- Dateien an Entire Output Management (NOM) übergeben.

## Benutzung von Ressourcen

---

Ressourcen können entweder real existierende Ressourcen darstellen oder aber fiktiv sein. Die Existenz realer Ressourcen kann mit Hilfe von Entire System Server Funktionalität festgestellt werden. Zum Beispiel können Sie den auf jeder verfügbaren Platte vorhandenen Freiplatz bestimmen, das Vorhandensein einer beliebigen katalogisierten Datei oder die aktuelle Anzahl gerade laufender Jobs.

Dem gegenüber stehen die Ressourcen, die nur innerhalb von Entire Operations eine Bedeutung besitzen. Der System-Administrator kann für sie einen Anfangswert bestimmen, der Anwender kann dann diese Ressource zur Regulierung des Job-Flusses benutzen.

Definiert er zum Beispiel eine bestimmte Menge dieser Ressource als Voraussetzung eines Jobs, wird dieser Job solange nicht ausgeführt, bis die geforderte Menge zur Verfügung steht. Der Anwender kann also eine Kombination aus Ressourcen benutzen, um die gewünschte Reihenfolge der Jobdurchführung innerhalb eines Job-Netzwerks zu erreichen oder um die parallele Ausführung von Jobs zu verhindern.

Weitere Einzelheiten entnehmen Sie dem Kapitel **Resources** in der *Entire Operations Administration Documentation* und dem Kapitel **Ressourcen** im *Entire Operations Benutzerhandbuch*.

## Dynamische Generierung von Job-Steueranweisungen (JCL)

---

Bei der Definition eines Jobs innerhalb eines Job-Netzwerks kann der Anwender bestimmen, daß dessen Job-Steueranweisungen dynamisch generiert werden, und zwar entweder zur Zeit der Jobaktivierung oder der Jobdurchführung. Die dynamische Generierung von Job-Steueranweisungen ist mit Hilfe der Entire Operations MACRO Funktionalität realisiert, als Erweiterung der Sprache Natural.

Ein solches MACRO besteht aus normalen Natural-Anweisungen sowie Job-Steueranweisungen beliebigen Formats. Innerhalb der Job-Steueranweisungen können durch Fluchtsymbole gekennzeichnete Variablen verwendet werden, die während der dynamischen Generierung durch ihre Werte ersetzt werden.

Diese Variablenwerte werden aus den benutzer-definierten Symbol-Tabellen entnommen, in denen den Symbolen die aktuellen Variablenwerte zugeordnet sind. Welche Symbol-Tabelle benutzt werden soll, wird bei der Definition von Jobs und Job-Netzwerken angegeben. Zum Zeitpunkt der Variablenersetzung (entweder Aktivierung oder Ausführung) wird jedes in den Job-Steueranweisungen referenzierte Symbol, das nicht in der Symbol-Tabelle des Jobs gefunden wird, in den Symboltabellen des Eigentümers SYSDBA gesucht. Die Anzahl der Einträge in einer Symbol-Tabelle ist beliebig, die Anzahl der Symbol-Tabellen ebenfalls.

Darüberhinaus stellt Entire Operations eine Reihe von Standardvariablen dem dynamisch zu generierenden Programm in einem Parameterabschnitt zur Verfügung, wie z.B. den Namen des Job-Eigentümers, des Jobs, des Job-Netzwerks und des ursprünglichen Zeitplan-Datums. Ebenso sind Natural-Systemvariablen wie z.B. DATE, TIME, USER verfügbar. Da all diese Parameter an beliebigen Stellen der Job-Steueranweisungen eingebaut werden können, können unterschiedliche Folgen der Job-Steueranweisungen in Abhängigkeit von DATE, TIME, usw. generiert werden.

Informationen zum Editieren von Jobs des Typs MAC entnehmen Sie dem Abschnitt **Editing JCL von MAC (Macro)-Job editieren** im Kapitel **Job-Verwaltung** in Ihrer *Entire Operations Benutzerdokumentation*.

Die Möglichkeit zur dynamischen Generierung von Job-Steueranweisungen bietet Entire Operations auf allen unterstützten Plattformen an (OS/390, VSE/ESA, BS2000/OSD und UNIX).

## Beispiel 1: Dynamische Job-Steueranweisungen in einer OS/390-Umgebung:

Die Symbol-Tabelle des MACRO-Programms soll wie folgt aussehen:

Symbol Name	Aktueller Wert
BIBLIOTHEK	SN.SYSF.SOURCE
KLASSE	G

Eine der Variablen im übergebenen Parameterabschnitt soll den Wert haben:

P-OWNER	NET1
---------	------

System-Variablen sollen die folgenden Werte besitzen:

*TPSYS	COMPLETE
*DEVICE	BATCH
*INIT-USER	SN

Die folgende Auflistung stellt nun ein Natural MACRO-Programm dar, daß einen Parameterabschnitt und Job-Steueranweisungen umfaßt. Variablen-Namen sind durch# gekennzeichnet und werden aus obiger Symbol-Tabelle entnommen:

```
# DEFINE DATA PARAMETER USING NOPXPL-A
# LOCAL          /* MUSS SO EINGEGEBEN WERDEN
# END-DEFINE
//SNMAC4 JOB ,#P-OWNER,MSGCLASS=X,CLASS=#KLASSE
//STEP01 EXEC PGM=NOPCONTI,PARM='C0004'
//STEPLIB DD DISP=SHR,DSN=#BIBLIOTHEK
/* DEVICE: *DEVICE, INIT-USER: *INIT-USER
/* TPSYS: *TPSYS
# IF #KLASSE = 'G'
/* DIE MSGCLASS IST TATSAECHLICH 'G'
# ELSE
/* ES WURDE EINE ANDERE MSG-CLASS GEFUNDEN
# END-IF
/*
```

Die hieraus dynamisch generierten Job-Steueranweisungen lauten:

```
//SNMAC4 JOB ,NET1,MSGCLASS=X,CLASS=G
//STEP01 EXEC PGM=NOPCONTI,PARM='C0004'
//STEPLIB DD DISP=SHR,DSN=SN.SYSF.SOURCE
/* DEVICE: BATCH, INIT-USER: SN
/* TPSYS: COMPLETE
/* DIE MSGCLASS IST TATSAECHLICH 'G'
/*
```

## Beispiel 2: Dynamische Job-Steueranweisungen in einer BS2000/OSD-Umgebung:

Die Felder des View DB-INFO sollen hierbei nach dem Datenbank-Zugriff (FIND Anweisung) folgende Werte haben:

Feld	Wert
NUCLEUS	055
LP1	1000
NU1	100
ACCOUNT	EXAMPLE
NH1	4000
MSG	FHL
VERSION	524

Die Variablen aus dem Parameterabschnitt sollen folgende Werte haben:

Variable	Wert
P-OWNER	OS
P-JOB	NUC055
P-EXECUTION-NODE	055

Es wird keine Symbol-Tabelle für den MACRO-Job referenziert, alle Variablen sind durch vorangestelltes Fluchtsymbol (hier: #) gekennzeichnet:

```
# DEFINE DATA PARAMETER USING NOPXPL-A
# 1 L-JOB
# 1 REDEFINE L-JOB
# 2 L-JOB-A (A3)
# 2 L-JOB-NUC (N3)
# LOCAL /* LOKALE VARIABLEN KOENNEN HIER EINGEFUEGT WERDEN
# 1 DB-INFO VIEW OF DB-INFO
# 2 NUCLEUS
# 2 LP1
# 2 NU1
# 2 ACCOUNT
```

```

# 2 NH1
# 2 MSG
# 2 VERSION /* Z.B. 524
# 1 LWP (N7)
# 1 NUC (N3)
# 1 SPOOL (A10) INIT <'NOSPOOL'>
# END-DEFINE
# *
# MOVE P-JOB TO L-JOB-A
# MOVE P-EXECUTION-NODE TO NUC
# F1. FIND DB-INFO WITH NUCLEUS = NUC
/.NUC NUC LOGON #P-OWNER,#ACCOUNT
/OPTION MSG=#MSG
/REMARK
/REMARK NUCLEUS #NUC
/REMARK
/SYSFILE SYSLST = NUC NUC..LST.NUC
/SYSFILE SYSDTA = SYSCMD
/FILE ADA VERSION..MOD, LINK=DDLIB
/FILE *DUMMY, LINK=DDLOG
/FILE *DUMMY, LINK=DDSIBA
/FILE ADA NUC..ASSO, LINK=DDASSOR1, SHARUPD=YES
/FILE ADA NUC..DATA, LINK=DDDATAR1, SHARUPD=YES
/FILE ADA NUC..WORK, LINK=DDWORKR1, SHARUPD=YES
/EXEC (ADARUN, ADA VERSION..MOD)
# COMPUTE LWP = F1.LP1 * (F1.NU1 + 100)
ADARUN PROG=ADANUC, LP=F1.LP1, LU=65535, LWP=#LWP ADARUN
DB=#NUC, NU=#NU1, NC=20, TT=600, TNAE=1800 ADARUN NH= NH1
/SYSFILE SYSLST = (PRIMARY)
/SYSFILE SYSDTA = (PRIMARY)
/SYSFILE SYSOUT = (PRIMARY)
/LOGOFF SPOOL
# END-FIND

```

Hieraus resultieren folgende dynamisch generierte Job-Steueranweisungen:

```
/.NUC055 LOGON OS,EXAMPLE
/OPTION MSG=FHL
/REMARK
/REMARK      NUCLEUS 055
/REMARK
/SYSFILE SYSLST = NUC055.LST.NUC
/SYSFILE SYSDTA = SYSCMD
/FILE ADA524.MOD, LINK=DDLIB
/FILE *DUMMY, LINK=DDLOG
/FILE *DUMMY, LINK=DDSIBA
/FILE ADA055.ASSO, LINK=DDASSOR1, SHARUPD=YES
/FILE ADA055.DATA, LINK=DDDATAR1, SHARUPD=YES
/FILE ADA055.WORK, LINK=DDWORKR1, SHARUPD=YES
/EXEC (ADARUN, ADA524.MOD)
ADARUN PROG=ADANUC, LP=1000, LU=65535, LWP=200000 ADARUN
DB=055, NU=100, NC=20, TT=600, TNAE=1800 ADARUN NH=4000
/SYSFILE SYSLST = (PRIMARY)
/SYSFILE SYSDTA = (PRIMARY)
/SYSFILE SYSOUT = (PRIMARY)
/LOGOFF NOSPOOL
```

### Beispiel 3: Dynamische Job-Steueranweisungen in einer UNIX-Umgebung:

Im folgenden Beispiel soll eine dynamische Symbol-Ersetzung innerhalb eine Bourne-Shell-Skripts erfolgen (als Fluchtzeichen dient hier \$):

```
#
# Bourne-Shell-Skript zur Überprüfung der Anzahl eingetragener
# Benutzer in /etc/passwd.
# Wenn mehr als $USER-LIMIT Einträge vorhanden sind,
# so wird das Skript mit exit 1 beendet.
#
#!/bin/sh
set -x
USER_COUNT=`wc -l < /etc/passwd`
echo Anzahl User auf Knoten `hostname` : $USER_COUNT
if test $USER_COUNT -gt $USER-LIMIT
then
    echo USER_COUNT_WARN
    exit 1
else
    echo USER_COUNT_OK
fi
```

Die zu verwendende Symbol-Tabelle soll wie folgt aussehen:

Symbol Name	Aktueller Wert
USER-LIMIT	100

Daraus ergibt sich die folgende ausführbare Shell-Prozedur:

```
#
# Bourne-Shell-Skript zur Überprüfung der Anzahl eingetragener
# Benutzer in /etc/passwd.
# Wenn mehr als 100 Einträge vorhanden sind,
# so wird das Skript mit exit 1 beendet.
#
#!/bin/sh
set -x
USER_COUNT=`wc -l < /etc/passwd`
echo Anzahl User auf Knoten 'hostname' : $USER_COUNT
if test $USER_COUNT -gt 100
then
    echo USER_COUNT_WARN
    exit 1
else
    echo USER_COUNT_OK
fi
```

*Anmerkung:*

*Alle zur Aktivierungszeit durch die Natural MACRO-Funktionalität aufgebauten Job-Steueranweisungen können durch den Anwender noch solange modifiziert werden, bis der Job tatsächlich gestartet wird. Natürlich sind diese Modifikationen dann nur für den aktuellen Netzwerklauf gültig.*

## System-Objekte editieren

---

Mit der Editor-Funktionalität in Entire Operations kann der Anwender folgende Objekte erstellen, anzeigen oder editieren:

- die Job-Steueranweisungen von Jobs, entweder aus der Master-Datenbank (und somit beliebigen externen Speicherquellen) oder aus der aktiven Datenbank. Änderungen an den Job-Steueranweisungen von aktiven Jobs beeinflussen nur den aktuellen Netzwerklaf, sie haben keine Auswirkungen auf die Master-Datenbank
- Natural-Programme und Benutzerrouitinen (Einzelheiten hierzu siehe Abschnitt **JCL oder Natural-Programme editieren** im Kapitel **Job-Verwaltung** in Ihrer *Entire Operations Benutzerdokumentation*);
- Entire Operations MAC (Makro)-Jobs (Einzelheiten hierzu siehe Abschnitt **JCL von MAC (Makro)-Jobs editieren** im Kapitel **Job-Verwaltung** in Ihrer *Entire Operations Benutzerdokumentation*);
- Online Dokumentation von Netzwerken, Jobs oder Ereignissen innerhalb von Jobs (Notizzettel-Information);
- Job-Protokolle (nur zum Anzeigen)
- Job-Ausgabelisten (nur zum Anzeigen).

Der Anwender von Entire Operations ist somit zum Beispiel in der Lage, so unterschiedliche Daten wie UNIX-Prozeduren, CA-LIBRARIAN-Dateien oder LMS-Dateien mit einem einzigen Editor zu bearbeiten.

Der Anwender kann den Editor aufrufen, indem er die Edit-Option im Verwaltungsbildschirm des entsprechenden Objekts auswählt. Der Editor stellt eine umfassende Funktionalität OS/390/ISPF-ähnlicher Befehle zur Verfügung. Diese sind dem Typ des Objekts angepaßt, welches editiert werden soll: zum Beispiel wird das Befehl **STOW** ein Natural Programm sichern und katalogisieren, es sollte aber für Natural MACRO-Programme nicht verwendet werden. Ein solches Natural MACRO-Programm wird gesichert, vorverarbeitet und katalogisiert durch Verwendung des **MACRO** Befehls.

Andere textverarbeitende Funktionen des Editors umfassen die Zentrierung, physische und logische Tabulator-Benutzung und Text-Überlagerung.

Umfassende online-Hilfe ist über eine PF-Taste innerhalb der Editor-Funktion verfügbar.

## Berichte generieren

---

Entire Operations bietet eine große Palette von Berichten an, um die Arbeit mit dem System auf allen Ebenen zu unterstützen. Der Anwender kann die Report-Funktionalität aufrufen, indem er die entsprechende Option aus dem Entire Operations *Hauptmenü* auswählt oder durch Eingabe des Direktbefehls **REPORT** auf einer beliebigen Bildschirmmaske. Berichte decken folgende Bereiche ab:

- Information über alle Jobs, selektierbar nach abgebrochenen, beendeten oder nicht gestarteten Jobs. Datumsintervalle sowie Netzwerknamen können zur weiteren Einschränkung der Suche angegeben werden. Die für diese Objekte erstellten Berichte können alle Ereignisse, Aktivierungszeit, Meldungen, Jobende-Status usw. enthalten. Alle Job-Auswertungen sind nach Protokollierungszeitpunkten sortiert.
- Netzwerk-Information, entweder in der Form eines kurzen Überblicks oder als umfassender Report, der detaillierte Informationen aller Netzwerk-Komponenten enthält. In allen Netzwerk-Reports sind Informationen über das Netzwerk selbst, alle Jobs, die Eingabe-Bedingungen und Ressourcen sowie bei Jobende-Behandlung inklusive der Ausgabe-Bedingungen enthalten. Diese ausführlichere Beschreibung beinhaltet zudem auch alle Prosa-Beschreibungen auf Netzwerk-, Job- und Ereignis-Ebene.
- Zeitplanübersicht für ausgewählte oder alle Netzwerke, inklusive der Liste aller Jobs, die innerhalb eines bestimmten Zeitfensters eingeplant sind. Diese Auswertung kann entweder für bereits vergangene Perioden angefordert werden, um z.B. alle nicht-erfolgten Netzwerk-Aktivierungen aufgelistet zu bekommen, oder aber für zukünftige Produktionsperioden, um Informationen für deren Vorhersage und Planung zu gewinnen.

Alle Berichte sind online verfügbar, die Daten können aber auch gedruckt werden. Hiermit ist eine einfache Möglichkeit einer Langzeit-Dokumentation gegeben.

Mit Hilfe des Entire Operations Import/Export-Werkzeugs kann der Inhalt der Master-Datenbank in eine flache Datei entladen werden. Neben der eigentlichen Verwendung für Datenmigration und Transport können Sie diese Funktion auch dazu benutzen, Ihr eigenes Berichtssystem aufzubauen.



## ANHANG

Weitere Informationen zum Thema Entire Operations finden Sie in folgenden Handbüchern:

*Entire Operations Konzept und Leistungsumfang*

*Entire Operations Benutzerhandbuch*

*Entire Operations Referenzhandbuch*

*Entire Operations Meldungen und Codes*

*Entire Operations Installation Documentation (englisch)*

*Entire Operations Administration Documentation (englisch)*





## **Entire Operations Konzept und Leistungsumfang**



## **Entire Operations Konzept und Leistungsumfang**